

協調的視覚数理モデル構築のための開発基盤
「HI-brain」の構築

占部 一輝

電気通信大学 大学院 情報システム学研究科
博士（工学）学位申請論文

2017年3月

協調的視覚数理モデル構築のための開発基盤 「HI-brain」の構築

主査	佐藤 俊治	准教授
委員	阪口 豊	教授
委員	工藤 俊亮	准教授
委員	末廣 尚士	教授
委員	吉永 努	教授

著作権所有者

占部 一輝

2017 年

協調的視覚数理モデル構築のための開発基盤「HI-brain」の構築

占部 一輝

概要

本研究の目的は、複雑システムの典型例である視覚の数理モデルを構築するためのソフトウェア基盤構築である。まずは基盤に必要となる機能と既存基盤の精査を行った。その結果、ロボットシステム開発を目的とした OpenRTM-aist がそれらを満たしうる拡張性を持つため、これをベースとしたソフトウェア基盤の構築を行った。必要機能の追加開発をいくつか行い、それらの外部評価として、「RT ミドルウェアコンテスト 2014」での「日本ロボット工業会賞」および「ベストサポート賞」、またロボットビジネス推進協議会が主催する「第 1 回 RT ミドルウェア普及貢献賞」の計 3 賞を受賞している。また本ソフトウェア基盤上で作成した数理モデルが、MIT Saliency benchmark test で世界 1 位を獲得している(2014 年 8 月 08 日)。

Collaborative software platform for computational models of vision: HI-brain

Kazuki Urabe

Abstract

The brain is a typical complex system that executes visual information analysis, motor control, selective allocation of memories, and so on. Brain researchers require to construct a numerical model which represent whole brain including these functions by connecting existing models, revising them and simulating them. The purpose of this study was develop a software platform to simulate the complex brain system numerically by computational models, especially focused on vision.

Examination and reuse of Existing platforms allow that the platform is constructed as less effort. The platform, HI-brain is focused on vision, especially functional-leveled such as image processing. First, I defined system requirements and examined whether the existing platforms meet the requirements. As the result, there is nothing that any platforms for brain research meet all requirements. Therefore, HI-brain for vision simulation based on RT-middleware and OpenRTM-aist, which is a software platform to develop robotic system.

A new datatype as a common interface of various vision models is provided. The new datatype and my software library enable automatic switching of transformation method of input/output data between vision models, i.e., shared memory or via computer network. I also provide a software package named by OpenCV-RTC which converts a lot of image processing functions of OpenCV into RT-components executable on OpenRTM-aist.

I show that novel models are efficiently developed on my platform; (1) connection between parvocellular and magnocellular layer of retinal ganglion cells and a V1 model, (2) revision of an existing model by taking apart into 9 pieces and permuting the one, (3) reuse and combination of existing models for estimation of fixation location of humans' eye, and (4) parallel/distributed computing using "Raspberry Pi."

内容

1.	はじめに.....	1
2.	モデルとソフトウェア基盤の調査.....	4
2.1.	脳機能の数理モデル化の粒度.....	4
2.2.	既存ソフトウェア基盤の調査.....	5
2.3.	代表的な疎粒度モデル.....	8
2.3.1.	Neocognitron.....	8
2.3.2.	Saliency map を用いた視覚的注視位置予測モデル.....	16
2.3.3.	Deep Neural Networks.....	21
2.3.4.	MT 細胞モデル.....	28
2.4.	まとめ.....	33
3.	HI-brain の開発方針.....	35
3.1.	プラットフォームの開発方針と要件.....	35
3.2.	既存ソフトウェア基盤と必要要件.....	43
3.3.	RT ミドルウェアと視覚数理モデルの親和性.....	45
4.	OpenRTM-aist の機能拡張.....	48
4.1.	視覚モデル用の共通入出力データ型.....	48
4.2.	複数モデルの同期実行機能.....	50
4.3.	データ通信の高速化.....	52

4.4.	コンポーネント化システム	56
4.5.	モデル用データベース	68
4.6.	データフォーマットの相互変換ライブラリ	70
4.7.	容易な環境実装方法	72
4.8.	まとめ	74
5.	モデル構築の具体例と結果	76
5.1.	器官モデルと脳機能モデル間の結合	77
5.2.	新規性の高い数理モデルの構築	79
5.2.1.	注視モデルの概要	79
5.2.2.	既存モデルの再利用によるモデル構築	81
5.2.3.	新規注視モデルと定量評価	83
5.2.4.	結果の考察	85
5.3.	既存速度選択性モデルの改良	86
5.4.	並列分散処理による大規模モデル実装例	98
5.5.	異なる環境で実装されたモデルの結合例	104
6.	まとめと考察	108
7.	参考文献	113
	関連論文の印刷公表の方法及び時期	i

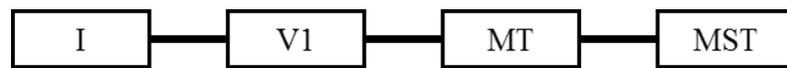
謝辭 iii

1. はじめに

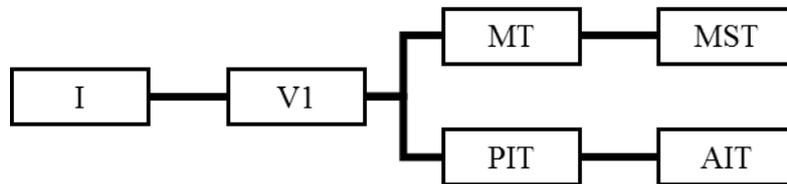
ヒトは様々な情報処理を日常的かつ無意識に行っている。例えば視覚系では、感覚情報から色や動きの解析 [1]，両眼視による奥行 [2] や凹凸の計算 [3]，物体認識 [4] や顔認識 [5] が挙げられる。これらは膨大な脳細胞が複雑に結合しあうことで、複雑な計算による結果であると解釈することができる。また視覚は単なる静的な画像処理を実行しているのではなく、入射光量に対する瞳孔変化や、処理対象を変化させるための眼球運動の制御などの運動制御も実行している。同様に、聴覚や触覚などの視覚以外のセンサー情報も視覚に影響を与える。このようにヒトは異種のセンシング、そしてセンシングした情報の複雑な処理、運動制御などを行う典型的な複合システムであるとみなすことができる [6]。

脳全体の計算処理の解明は、脳研究における重要な課題のひとつである。理想的には脳情報処理全体を記述する数理モデルを構築すればよい。しかし、脳は様々な機能を実現するために複雑な構成をしているため、単一の数理モデルで脳全体をモデル化することは容易ではない。この問題を回避するために、既存研究の多くは生理実験や心理実験ならびに対応する数理モデルを特定の細胞（ニューロン）や特定の脳機能に限定した研究が遂行されてきた。この結果、個々の実験結果を再現する数理モデルがこれまで多く提案されてきた。例えば、(i) 細胞レベルのモデルとしては、V1 野単純型細胞の入出力を記述する数理モデル [7]，(ii) 機能レベルのモデルでは、V1, MT, MST 野からなる

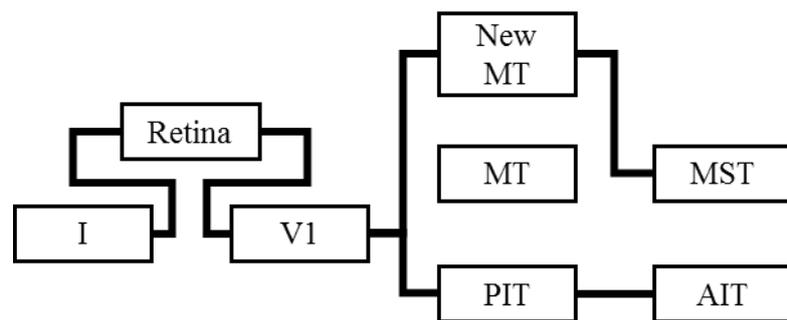
運動視 [8] (Fig. 1-A), 視覚的注意・注視位置推定のモデル [9] (Fig. 1-B), (iii) 器官レベルのモデルとして眼球・網膜モデル [10] などが提案されている。これらモデルが脳の一部を記述した部分的な数理モデルであり, 全体を記述したモデルではないことはすでに述べた。前述した脳研究の課題である「脳全体の計算処理の解明」を行うためには, Fig. 1-A や Fig. 1-B のように, 部分的な数理モデルを結合させ, より大規模な領域を記述したモデルを構築する必要がある。また現在においても様々な実験結果や知見が報告されているため, 新しい知見を反映させた新規モデルの開発も各種実験と並行して行う必要がある。一般的に既存モデルは, 新しい知見を説明できない場合が多い。したがって既存モデルの構成要素の一部を新しい要素に置換したり, 新しい機能や演算を挿入したりすることで, 新知見を再現・説明するモデルとなるように改良する必要がある (Fig. 1-C).



(A) 運動知覚に関する領野の直列結合



(B) 視覚的注視モデルの並列結合



(C) モデルの追加・置換によるモデル改良

Fig. 1: 機能レベルのモデルが再現する脳領野・器官の結合関係. 四角は脳領野や器官を表し, 線は領野間に情報の流れがあることを意味する. I から与えられた情報がそれぞれの領野にわたり処理される.

2. モデルとソフトウェア基盤の調査

脳数理モデル研究のためのソフトウェア基盤の開発をおこなうためには、対象となる既存数理モデルを深く理解し、モデル間の共通点を明らかにする必要がある。共通点を明らかにすることで、構築するソフトウェア基盤に必須となる機能や、開発コンセプトなどが明確になる。しかし脳機能のモデル化には様々なモデルの粒度が存在するため、本研究では対象とするモデル粒度を限定することとした。

本章ではまず、ソフトウェア基盤が対象とするモデルの粒度について説明する。次いで、本研究で対象とする数理モデルをシミュレーションする既存ソフトウェア基盤の問題点を明らかにする。さらに、いくつかの代表的な視覚数理モデルを説明し、これらのモデルに共通する事項を詳細に調べる。

2.1. 脳機能の数理モデル化の粒度

脳の数理モデル化は理学的にも工学的にも重要な研究課題であるが、次に述べるような難しさがある。端的に言えば、「目的」に応じてモデル粒度が異なることである。ここでは多く存在する数理モデルを大きく2種類に分け、それぞれの数理モデルについて説明する。

i. 細粒度モデル

脳内には多種多様な神経細胞が存在し、その形態や細胞ネットワークも部位によって大きく異なる。そこで、神経細胞の3次元形態や、電位や電流、

イオン流出入の変化を 1ms 程度のオーダーで忠実に再現するモデルが構築され、公開されている¹。創薬や病態予測などへの貢献が期待されている。

ii. 疎粒度モデル

構成要素単位を画像処理アルゴリズムとする数理モデルであり、種々の視覚領野（細胞集団）や視覚機能を工学的観点から考察できる。疎粒度モデルはヒトの脳の入出力特性を調べるような心理物理実験の結果を考察、モデル化する際に適しており [12]，工学的応用も期待できる。疎粒度モデルは脳領野や視覚機能を単位とする場合が多い（Fig. 1）。細胞モデルの入出力データは Hz や mV などの実数値である場合が多い。例えば細胞の出力である活動電位は、適当な時間窓で平均した平均発火頻度（Hz）で表現されることが多い。

次節では上記 i と ii に対応したソフトウェア基盤の有無や問題点について調査し、脳数理モデル研究を推進させるための方策について考察する。

2.2. 既存ソフトウェア基盤の調査

前節で述べた i と ii のモデル化に対応しているソフトウェアやシミュレータ環境を調査し、以降の開発に必要な要件を見出す。本節では最終的に、ii. 疎粒度モデルを

¹ <https://senselab.med.yale.edu/modeldb/>

記述するソフトウェア基盤開発が新たに必要であることを示す。

i. 細粒度モデルのためのソフトウェア基盤

NEURON² や GENESIS³ などのシミュレータが提案されており 2015 年現在も開発が進められている。これらの環境で構築されたモデルはデータベース⁴に登録されており、複数モデルの結合や共有化、ならびに再現性確保の促進が図られている。

ただし、これらのシミュレータは独自の記述言語やデータフォーマットを採用しているため、異なるシミュレータで構築されたモデル同士の相互結合や変換を困難にしている。この問題を解決するために MUSIC⁵と呼ばれるソフトウェアライブラリが提案されている。MUSIC は異なるシミュレータ間のデータ変換やシミュレータ間のデータ通信を行うことで、互換性を向上させるものである。しかしながら MUSIC を用いて具体的に新規モデルが構築された例は著者の知る限り存在しない。

ii. 疎粒度モデルのためのソフトウェア基盤

疎粒度モデルは、数値解析ソフトウェアである MATLAB や画像処理ライ

² <http://www.neuron.yale.edu/neuron/>

³ <http://genesis-sim.org/>

⁴ <http://senselab.med.yale.edu/modeldb/>

⁵ <http://www.incf.org/activities/our-programs/modeling/music>

ブラリである OpenCV⁶ を用いて実装されることが多い。前節の ii で示したように、疎粒度のモデルは数学的に良く知られた関数や信号画像処理フィルタなどを要素として記述されることが多いため、用いるソフトウェア基盤にはこれらの数学的要素や基本処理がプリインストールされていることが望ましい。加えて、新しい演算も容易に記述できることが望ましい。MATLABなどで記述された数理モデルを共有化する試みも精力的になされている。

しかし、モデルの入出力データフォーマットやパラメータ記述形式は研究者ごとに異なるため、複数モデルの相互結合とシミュレーションを行うことが難しい。事実、データベースに登録された疎粒度モデルの相互結合を行うためにはソースコードレベルの解読や解析が要求される。

以上の調査結果をまとめる。細粒度モデルでは独自フォーマットを使用しているなどの問題点はあるが、シミュレーション環境の整備や共有化、再現性の保証を目指した取り組みが進められている。一方、疎粒度モデルではモデルデータベースは存在しているものの、モデルの構築、結合、シミュレーションを可能とするソフトウェア基盤の整備が進んでいない。そこで本研究で作成する開発基盤（「HI-brain」と呼ぶ）は画像処理レベルのモデル（疎粒度モデル）をシミュレーション対象とすることとした。

⁶ <http://opencv.org/>

2.3. 代表的な疎粒度モデル

提案されている視覚モデルについていくつか例を挙げ、疎粒度の数理モデルに共通する要素や構造を説明する。汎用性の高いソフトウェア基盤を構築するためには、モデリング対象や目的が異なる疎粒度数理モデルを調査することが重要である。そこで本研究では数ある数理モデルの基礎となっている数理モデル 4 種を抽出した。具体的には (1) 文字認識モデルとして提案された Neocognitron [13] , (2) Itti と Koch による視覚的注意モデル [14] , (3) Deep Neural Networks, (4) Nishimoto と Gallant によるオプティカルフロー計算モデル [15] である。これらのモデルはそれぞれ, (1) Deep Neural Networks として 2016 年現在, 精力的に研究が進められている神経回路網モデルの基礎, (2) ヒトの視覚的注意特性を説明する数理モデル, (3) 2016 年現在様々な問題を解くために利用されている回路網モデル, (4) 実際の視覚形階層構造を取り入れた, 電気生理実験を説明する数理モデルである。

2.3.1. Neocognitron

ヒトは文章を読む際に, 個々の文字を認識することで文章全体の意味を理解することができる。網膜上に投影される文字の画像情報は, 環境 (暗所・明所) や形状の違い (筆跡・文字フォントの違い, 拡大縮小・位置ずれ・歪み) によって大きく変化するが, ヒトはこれらの違いに影響されない頑健な文字認識を可能にしている。このような様々な画像情報の変化に対して頑健に文字認識を実現したモデルの一つがネオコグニトロン

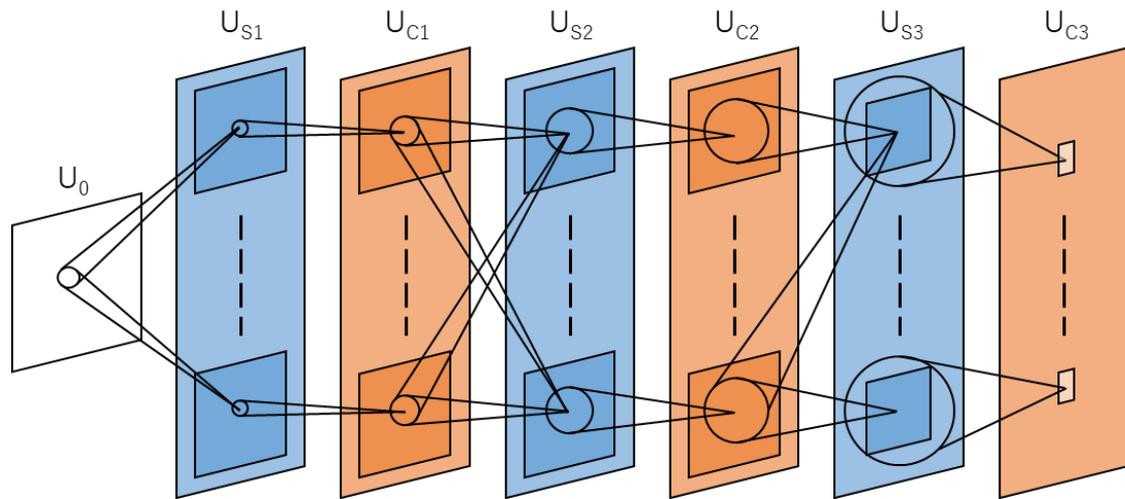


Fig. 2: Neocognitron の階層図. Neocognitron は複数の細胞層で構成されており, 各四角が細胞層を表している, 白の層(U_0), 青い層(U_0), 橙色の層(U_0)はそれぞれ入力層, S 細胞層, C 細胞層を表している. 各層内の四角は細胞面を表している. 最終層(U_{C3})の細胞面の反応によって, 与えられた入力 of 識別結果がわかる.

である [13]. ネオコグニトロンは学習機能を持った人工神経回路網であることも特徴の一つである.

ネオコグニトロンは複数の細胞層から構成されている (Fig. 2). U_0 は入力層であり呈示したパターンが与えられる. 細胞層は入力層を除けば S 細胞層と C 細胞層から構成されており, 図ではそれぞれ U_{S_l} と U_{C_l} として示している. l は層の階層を意味し, S 細胞層と C 細胞層は交互に階層的に配置されている. また各層内の四角形は神経細胞モデルが二次元的に配置されている細胞面を表している. 一つの細胞層は複数の細胞面で構成される.

S 細胞と C 細胞はそれぞれ大脳視覚野の単純型細胞と複雑型細胞と類似した反応特

性を示すように設計されている。S 細胞は主にパターン内の特徴を抽出しており、S 細胞が受け持つ領域内（受容野）に特定の特徴が呈示されると強く反応する。同一の細胞面内の S 細胞は同一の結合強度を持ち、異なる領域と結合している。そのため細胞面内の S 細胞はそれぞれ異なる領域から、同一の特徴を抽出する。C 細胞では呈示されたパターンの変化を許容するために、S 細胞による特徴検出結果を空間的にぼかし、さらにダウンサンプリングする処理を行っている。

S 細胞は入力層や C 細胞層から多数の入力を受け取っているが、これら結合の強度は学習によって変化させることで認識率を向上させている。すなわち、S 細胞が抽出する画像特徴は、学習時のパターンや学習方法に依存して変化する。C 細胞は複数の S 細胞からの固定された結合を受け取っており、その結合強度は固定である。

細胞層は層を重ねるにつれ、より広範囲の入力から情報を得るようになり、S 細胞が抽出する画像特徴も複雑になる。たとえば S 細胞の場合は低層では簡単な特徴（特定の方位の線など）を抽出し、高層の S 細胞はより複雑な特徴（学習パターンの一部など）を検出するようになる。最後の C 細胞層では各細胞面内の細胞が一つとなり、入力層全体から入力を受け、入力パターン全体の情報を統合している。入力パターンのカテゴリに該当する C 細胞一つのみが反応することで、与えられたパターンの識別を行っている。

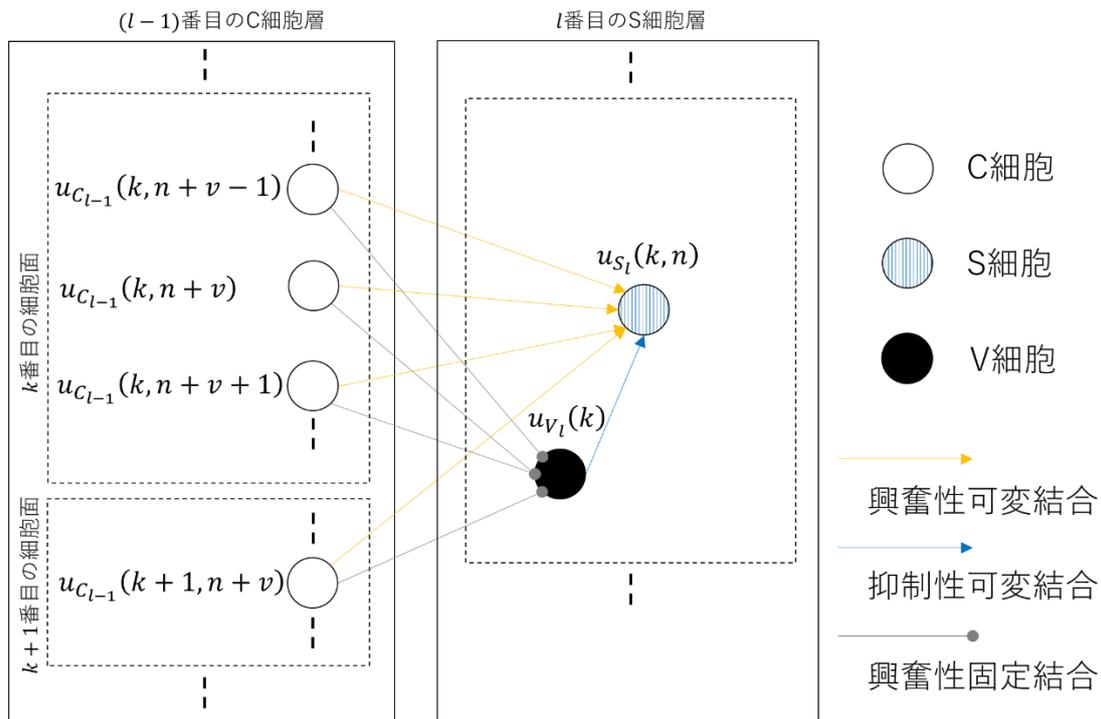


Fig. 3: C 細胞層から S 細胞層への結合図. 白丸, 青丸, 黒丸はそれぞれ C 細胞, S 細胞, V 細胞を表している. S 細胞は C 細胞層の全細胞面から入力を受け取っており, これらは S 細胞と受け持つ領域を共有している. C 細胞からの結合は興奮性であり, S 細胞の活動を促進する. それに対して, V 細胞との結合は抑制性で, これは活動を抑制する効果がある. これら S 細胞に対する結合は学習過程において変化する.

C 細胞層と S 細胞層間の結合関係は Fig. 3 のようになる. S 細胞, C 細胞, V 細胞の出力はそれぞれ $u_{S_l}(k, n)$, $u_{C_{l-1}}(k, n+v)$, $u_{V_l}(k)$ で表され, S 細胞は前層の C 細胞から興奮性の可変結合を介して入力を受け取り, V 細胞からは抑制性の可変結合を介して入力を受け取る. これらは学習結果に応じて, 結合強度が変化する. また V 細胞は S 細胞と同様に興奮性の入力を受け取るが, その結合は固定結合であり, 学習によって変化することはない. S 層に入力を渡す C 細胞は複数の細胞面にまたがっており, S 細胞が

受け持つ領域に関わる C 細胞から入力を受け取る。S 細胞は C 細胞からの入力 $u_{C_{l-1}}$ を複数受け取っており、これらが大きいほど S 細胞の出力もまた大きくなる。逆に V 細胞からの抑制性の入力を受け取る場合、S 細胞の出力が小さくなる。

S 細胞の出力を以下の式に示す：

$$u_{S_l}(k, n) = \varphi \left[\frac{1 + \sum_{k=1}^{K_{l-1}} \sum_{v \in R_l} w_{e_l}(k, v) u_{C_{l-1}}(k, n + v)}{1 + w_{i_l}(k) u_{V_l}(k)} - 1 \right] \quad (1)$$

w は結合強度を意味しており、興奮性の結合の場合は w_{e_l} 、抑制性の場合は w_{i_l} である。

K_{l-1} は C 細胞層の細胞面の数を、 R_l は S 細胞が受け持つ領域を意味する。C 細胞から与えられた入力 $u_{C_{l-1}}$ は w_{e_l} を用いて線形結合され、 w_{i_l} によって重み付けられた V 細胞からの入力による除算によって抑制される。関数 $\varphi[x]$ は以下の式となる。

$$\varphi[x] = \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases} \quad (2)$$

抑制細胞である V 細胞もまた S 細胞と同じ C 細胞から入力を受け取り、C 細胞からの入力の平均値(2乗平均)をその出力としている。すなわち

$$u_{V_l}(k) = \sqrt{\sum_{k=1}^{K_{l-1}} \sum_{v \in R_l} w_{f_l}(v) u_{C_{l-1}}^2(k, n + v)} \quad (3)$$

V 細胞は C 細胞からの抑制性固定結合を介して入力を受け取っている。この結合は固定であるため、結合強度である $w_{f_l}(v)$ は以下の式に従う。

$$\sum_{k=1}^{K_{l-1}} \sum_{v \in R_l} w_{f_l}(v) = 1 \quad (4)$$

ネオコグニトロンは、入力データに学習結果によってパターン認識機能を獲得する。

学習手法である「教師あり学習」と「教師なし学習」の両方がネオコグニトロンの強化に使用可能である。教師あり学習では入力データに対する出力結果と正規の答え(ラベル)を比較, その正誤から入力データの傾向を学習することで新しい入力に対して正しい出力を返すようになる。対して教師なし学習では, 与えられた入力データ自体からその傾向を学習し, 新しいデータの識別を可能にする。ここでは教師なし学習を使用した場合の, 学習過程について説明する。

ネオコグニトロンは, 初期状態からパターンの認識機能があるわけではなく, 学習過程においてその機能を決定する。具体的には学習過程において与えられる入力データに従い, S 細胞層への入力結合強度を変化させる。幼児が自然に言語を習得するように, 分類と認識機能を自ら獲得することを「自己組織化」といい, ネオコグニトロンの自己組織化は次の規則に従う。

1. ある細胞がその近傍領域内で最大の出力を出している場合, その細胞に与えられる入力が0であるものを除き, その入力結合は入力強度に比例して強化される
2. 1で強化される細胞と同一の細胞面に別の細胞が存在する場合, これら細胞も同様に強化される

この規則は S 細胞に対する興奮性・抑制性両方の可変入力結合を対象とする。

近傍領域内で最大の出力を出す細胞は, あたかも結晶成長での核のような働きをする

ため seed cell と呼ばれる。seed cell は S 層内のおよそ同じ位置に受容野を持つ S 細胞のグループ(ハイパーカラム)からひとつずつ選ばれる。ハイパーカラムにはすべての種類の特徴抽出細胞が含まれている。細胞面内の細胞がそれぞれ異なる領域から同一の特徴を抽出するのに対し、ハイパーカラム内の細胞は同一の領域から異なる特徴を抽出している。

学習パターンが呈示されたとき、ハイパーカラム内から最も強い出力を出す細胞が seed cell として選ばれる。その際、同一の細胞面に 2 つ以上の seed cell がある場合は出力の強い seed cell が選ばれる。また逆に細胞面に seed cell がない場合は、そのままのものとする。したがって、最大一つの seed cell が各細胞面から選ばれる。そして細胞面内の S 細胞は seed cell と同様に強化される。この時の更新則を以下に示す。

$$\begin{aligned}\Delta w_{e_l}(k, v) &= q \cdot w_{f_l}(v) \cdot u_{C_{l-1}}(k, n + v) \\ \Delta w_{i_l}(k) &= q \cdot u_{V_l}(k)\end{aligned}\tag{5}$$

q は更新速度を意味し、この値が大きいほど少ないステップで学習を終える。この更新は異なる学習パターンが与えられるたびに行われるため、学習パターンが与えられるたびに、通常異なる細胞が seed cell として選ばれる。結果的に細胞面内の S 細胞は seed cell と同様の結合重みをもつようになり、C 細胞との結合関係のみが異なるようになる。

福島らが公開しているプログラムの出力結果の一部を Fig. 4 に示す⁷。U₀ に与えた入

⁷ https://visiome.neuroinf.jp/modules/xoonips/detail.php?item_id=375

カパターンに対して、 U_{C1} では単純な特徴であるエッジの検出をおこない、高層のC細胞ではより複雑な特徴を検出している。最終層である U_{C4} では入力パターンに対応した一つの細胞が反応していることがわかる。

ネオコグニトロンは複数の種類が存在しており、目的に応じて構造や学習規則が異なる [16]。例えば、Fig. 4には本節で説明していない U_G の層があるが、これは網膜やLGNに対応した層である。網膜やLGNでは一般的に入力パターンのコントラストに対して選択的に反応する細胞が確認されており、 U_G はこれを模したものである。このようにネオコグニトロンはその構造の違いや学習規則の変更することで、様々な種類の入力パターンに対する認識率を上げている。そのため、新たなネオコグニトロンを構築する場合は、様々なネオコグニトロンの違いを理解し、改良し、実装する必要がある。しかし、ネオコグニトロンの違いや挙動を理解するには、論文を熟読し、実際に構築する必要があるが、これは非常に手間である。既存のネオコグニトロンに対して、最小限の変更で新たなモデルの構築が可能であるならば、これらの手間を減らすことも可能になると考えられる。

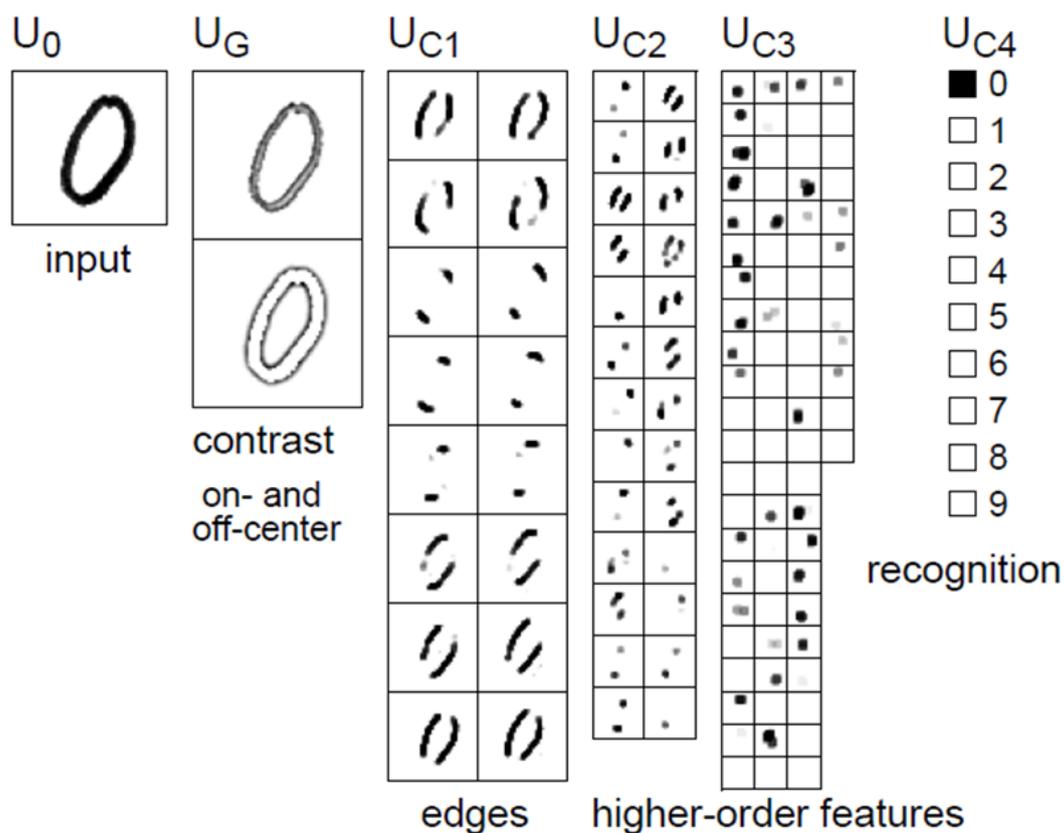


Fig. 4: Neocognitron の出力結果の一例. 最終的な出力結果を見ると, 入力層に与えられたパターンに対して'0'と認識していることがわかる.

福島らのプログラムは C++ のプログラムコードで書かれており, コード内の説明も少ないため, その改良は困難である. またこのプログラムは出力結果を PostScript 形式で出力するため, その結果に対して何らかの処理を加えることは難しい.

2.3.2. Saliency map を用いた視覚的注視位置予測モデル

ヒトは雑然とした情景内の中から目立つものや興味のあるオブジェクトに対して, 自然と視線を向ける. この性質は生物の進化的意義があり, 具体的には生物がその捕食対

象や非捕食対象を視野内で素早く検出する際などに必要とされるために獲得した機能だと考えられる。また、外界像から得られる情報は膨大であるため（視神経から得られる情報量はおよそ $10^7 \sim 10^8 \text{bit/s}$ ）、これらすべての情報に対して同一の処理を与えると処理時間が長くなり、処理に要するリソース（細胞数）も膨大になる。現実的には全視野内の特定の位置内に存在するオブジェクトに対してのみ、適切な処理を与える機能が必要になる。例えば車の運転中に、前に急停車したことを認識する場合は、前の車のみに対して注意を向け、それ以外のオブジェクトに対しては注意を向ける必要はない。視覚的注視はそのために必要な機能であり、膨大な情報から必要な情報の選択を行うことができる。

視覚的注視機能は大きく2つに分類される。一つは、視野内の目立つものに処理を集中させる「ボトムアップ」な視覚的注視、もう一つは、目標に処理を集中させる「トップダウン」な視覚的注視である。ボトムアップな視覚的注視は、視野内に注意を引き付ける特徴がある(ポップアウトする)場合に「自然」に注意を引き付けることを指す。逆にトップダウンな視覚的注視は、特定のオブジェクト(例えば、色と方位が異なるバー)に「自発的」に注意を向けることを指す。

現在様々な視覚的注視機能を記述した計算モデルが提案されているが、これらの多くは Fig. 5 に示す構成を基本としている。この図は Koch らによって提案されたボトムアップ型注視モデルの構成図である [14]。ボトムアップ型注視の場合、目立つ箇所つま

り顕著性(Saliency)の高い位置に注意や注視位置を集中させる傾向にある。また Saliency が高い領域は、色や方位などの特徴が他の領域と異なる傾向にある。具体的には次の処理が行われる。

- (1) 各特徴量から特徴量のマップを作成
- (2) 画像内の顕著性を記載した Saliency Map の作成
- (3) 注視位置の決定

まずは視野全体の情報に対して、色相(赤, 緑, 青, 黄など), 輝度(On, Off など), 方位(0° , 45° , 90° , 135° など)などの様々な画像特徴を様々な空間スケールで抽出する。その後異なるスケール間での差分を相関計算(Centre-surround differences)によって得ることで、個々の特徴における特徴マップを作成する。さらに、各特徴マップで一部の Saliency が突出している (マップの一部の Saliency が高く、それ以外は低い) 場合は強調, 突出していない (マップ全体の Saliency が高いまたは低い) 場合は抑制する。このような空間的競合を繰り返すことで、各特長マップで突出した特徴のみが残るようになる(Spatial competition)。これらを統合して一つの顕著度マップ(Saliency map)を作製する。これは Saliency を可視化したものであり、これを基に注視位置を決める必要がある。ボトムアップな視覚的注視では、視野内の目立つ (Saliency の高い) 箇所に注視位置が集中することは前にも述べた。そのために、一般的な視覚的注視モデルでは winner-take-all アルゴリズムが使用されており、最も Saliency が高い位置が他の個所の Saliency を

抑制することができる。ここまでが一つの注視位置を決める手順である。しかし、ヒトは一つの個所を目的なく注視し続けることはなく、その他の目立つ箇所に注視位置を変化させていく。逆に言えば、一度注意を向けた位置に対して再び注意を向けにくいことを意味しており、この性質は *Inhibition of return* と呼ばれている。この注視位置の変化をモデル化するために、一度注視した箇所の *Saliency* を抑制する機構が導入される。

生理学的には、網膜、上丘、外側膝状体、初期視覚皮質などでは視野内の簡単な特徴を検出している。より高次領野になるほど、角や交差 [17]、シェイプフロムシェーディング [18,19] などのより複雑な特徴に対して選択的になる。Fig. 5 で示した *Saliency* マップモデルの構造は初期視覚野で得られるような、色相や方位などの簡単な情報のみで構成されているが、高次領域で得られるような特徴に関しては考慮していない。そのためより正確な視覚的注視モデルを構築するには、複雑な特徴の検出とその特徴マップの作製しなければならない。また前述のモデル構造は、ボトムアップ型の視覚的注視のみを記述しており、トップダウン型については考慮していない。トップダウン型も組み込むことを考えると、より複雑な特徴の識別機能を追加する必要がある。

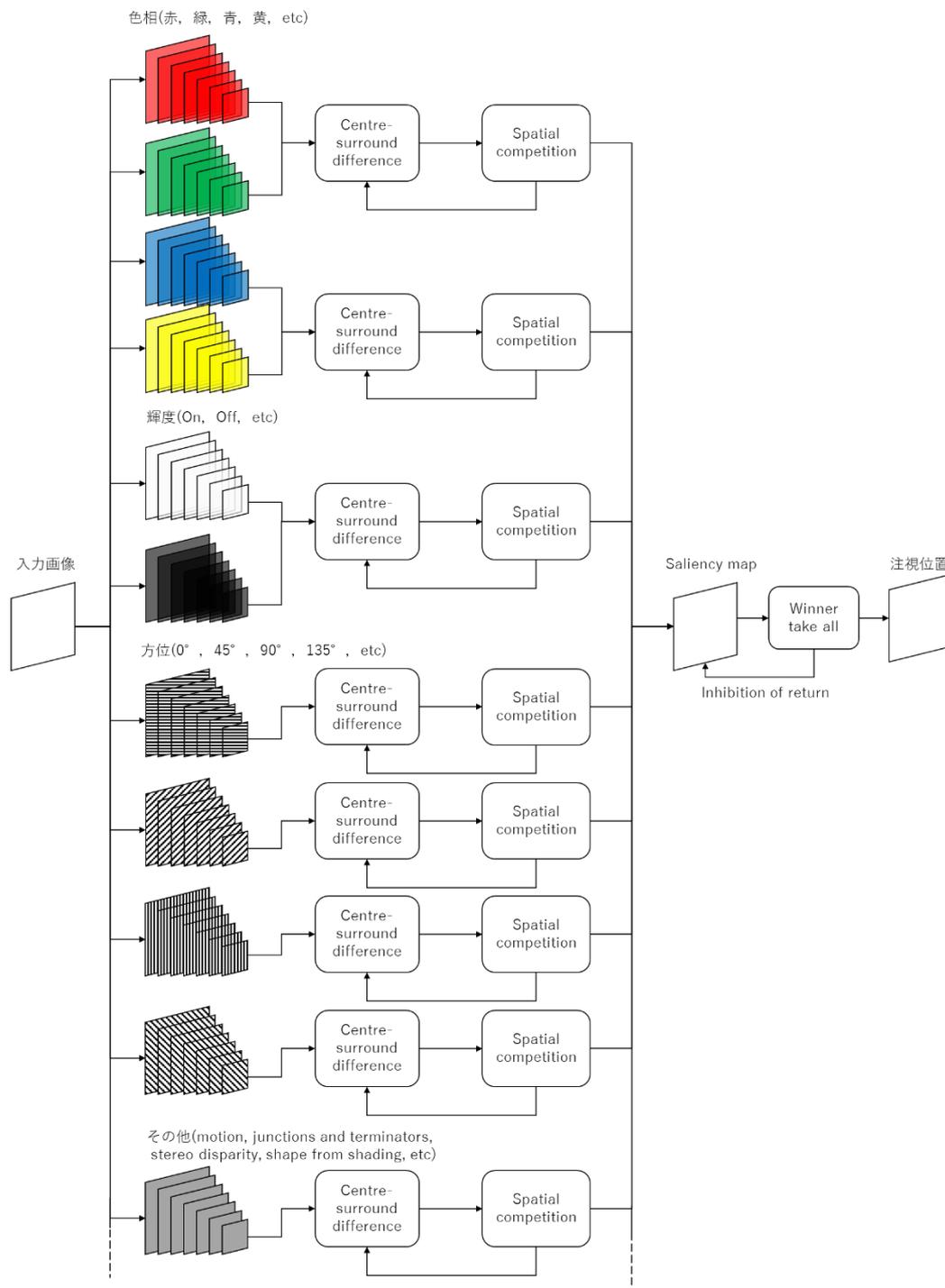


Fig. 5: Koch らが提案する視覚的注視モデルの構成図. 入力画像に対して色相・輝度・方位など異なる特徴を異なるスケール(画像サイズ)で抽出, スケール間の相関を計算 (Centre-surround difference), 空間的競争(Spatial competition)によって Saliency map を作成. Saliency の高い箇所を注視位置とする.

2.3.3. Deep Neural Networks

網膜から与えられた視覚情報は様々な脳領野を経由することで、より複雑な視覚情報処理が施される。一般的には経路する脳領野が増えるほど、より複雑な機能が構築される。初期視覚機能では物体のエッジやその傾きなどの単純な特徴を抽出しているが、処理が進むにつれエッジに組み合わせによる角や交差の情報を検出、より深い領域の機能では顔や文字などのオブジェクトの情報を検出するようになる。単体では単純な処理をするニューロンを複雑に組み合わせることによって、脳全体として非常に複雑な処理をしていると考えることができる。前述した Neocognitron も、低層で抽出した特徴から、高層になるにつれてより複雑な特徴を認識するようになり、最終的には文字の識別を行うように構築されている。

より膨大な数のニューロンとそれら結合を持つ複雑なニューラルネットワークを構築し、膨大な入力データを基に学習すれば、より高次のオブジェクトを知覚することができることが期待される。Leらはこれを実際に行い、ヒトの顔や体、猫の顔などの物体をヒトに教えられることなく、識別するニューロンを構築した[20]。

この論文では、ラベル情報のない画像から、ヒトの顔などの高次特徴の検出器を構築することを目的としている。一般的に、「おばあさん細胞」と呼ばれる特定のオブジェクトを表現する仮想ニューロンの存在は以前から問われていた。このような細胞に関する調査はこれまで行われており、たとえば顔や手に選択的な細胞の研究 [21]、特定の

人物に選択的な細胞の研究などがある。この論文では「おばあさん細胞」の存在を示唆する結果が得られた。

このモデルではラベル情報を与えない、いわゆる教師なし学習によってネットワークを学習している。これまでの画像処理では、教師データ(ラベル)を含んだ画像データを必要とした。例えば、顔検出器を構築する場合はラベルと顔を含んだデータセットが必要となる。しかしラベルを持つ画像データはほとんど存在しない。故にラベルなしのデータを基にネットワークを学習しなければならない。しかし「顔」の概念は先天的にかつ、「顔」の概念を教示する教師の存在によって獲得されるものではないため、ラベルなし(教師なし)データからも学習できるのではないかと考えられる。そこで教師なし学習によって、高次機能の構築が可能であるかを調査した。

既存研究においても RBMs(Restricted Boltzmann machine)[22] やオートエンコーダ [23,24] , スパースコーディング [25] , K-means[26] などの様々な手法で、モデル構築を行ってきた。しかし既存モデルの多くは単純な特徴を検出するものである。より高次画像特徴を抽出するモデルを構築するためには、学習時に長い時間を要する [27] 。現実的な時間内に学習を終えるためにはこれまで、訓練データやネットワーク自体のサイズを減らすなどの工夫がおこなわれてきたが、このことが高次特徴の学習の弊害となっているとも考えられる。そのため、このモデルでは学習に使用するデータセット、モデル、計算リソースなどを大規模に拡張し問題の解決を試みている。

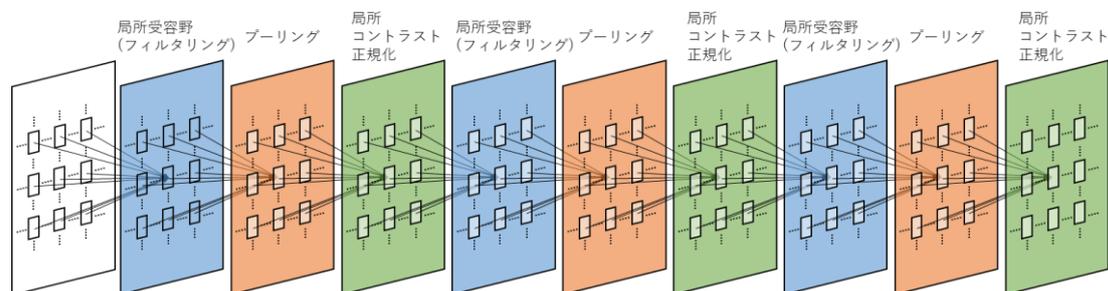


Fig. 7: ネットワーク全体の構造図. それぞれの四角が一つの層を, 層内の四角は層を構成するニューロンを表している. ネットワークは局所受容野(青), プーリング(橙), 局所コントラスト正規化(緑)の3つの要素で構成されている.

ネットワークのアルゴリズムは, 局所受容野, プーリング, 局所コントラスト正規化という3つの要素で構成されている. まず局所受容野によって, 入力の小領域に対してフィルタリング処理を施す. フィルタ自体が学習によって獲得される. 次に入力の変形に対する頑健性を持たせるために, 局所 L2 プーリング [29,32,33] と局所コントラスト正規化 [34] を適用する. 生理学的には, 局所受容野とプーリングは V1 野における単純型細胞と複雑型細胞の役割と相同である. 同様に局所コントラスト正規化は, 生理学的知見や一部モデルで行われている local subtractive and divisive normalization に対応する [34,35,36]. ネットワーク全体は, これら処理を3回繰り返した9層構造となる(Fig. 7). この構造と基本的な処理は前述の Neocognitron や HMAX [37,38,39] と同様であることに注意されたい.

このモデルを3日間, 1000台のマシンの組まれたクラスタ上で学習させた結果を Fig. 8, Fig. 9 に示す. これらはヒトの顔, 人体に選択的なニューロンが最も強く反応したテ

ストデータと、そのニューロンに最適な刺激を可視化したものである。強く反応したテストデータを見ると、それぞれ選択的なオブジェクトを多く含んでいることがわかる。また、数値最適化によって可視化した最適な刺激では、それぞれのオブジェクトを識別し、その概念を学習していることが確認できる。

膨大な数のニューロンとその結合によって構成されたニューラルネットワークを、多数の CPU を含む大規模な計算インフラと Youtube から取得した多数の大きい画像データを用いて学習した場合、そのネットワークは入力データに共通するオブジェクト（ヒトや猫などの高次特徴）の認識を自ら獲得するだろうか？この研究ではそれを実際に行い、それを確認することができた。入力データには、何が「ヒトの顔」で、どれが「猫の顔」などのラベルデータ含まれていない。しかし、ネットワーク内のニューロンの一つが各オブジェクトに強い反応を示すように学習した。

本ネットワークは 2016 年現在、最大のニューラルネットワークの一つであり、他のネットワーク [27, 40] がおよそ 1000 万の結合を持つのに対し、このネットワークは 10 億の結合を持つ。しかしヒトの視覚皮質と比較した場合は小規模であり、ヒトのニューロンとシナプスの数と比較すると、 10^6 倍異なる [41]。よりヒトに近いオブジェクト認識機能を構築するには、より大量の入力データ、ハードウェアの計算リソース、大規模なネットワークモデルの構築が必要となるだろう。

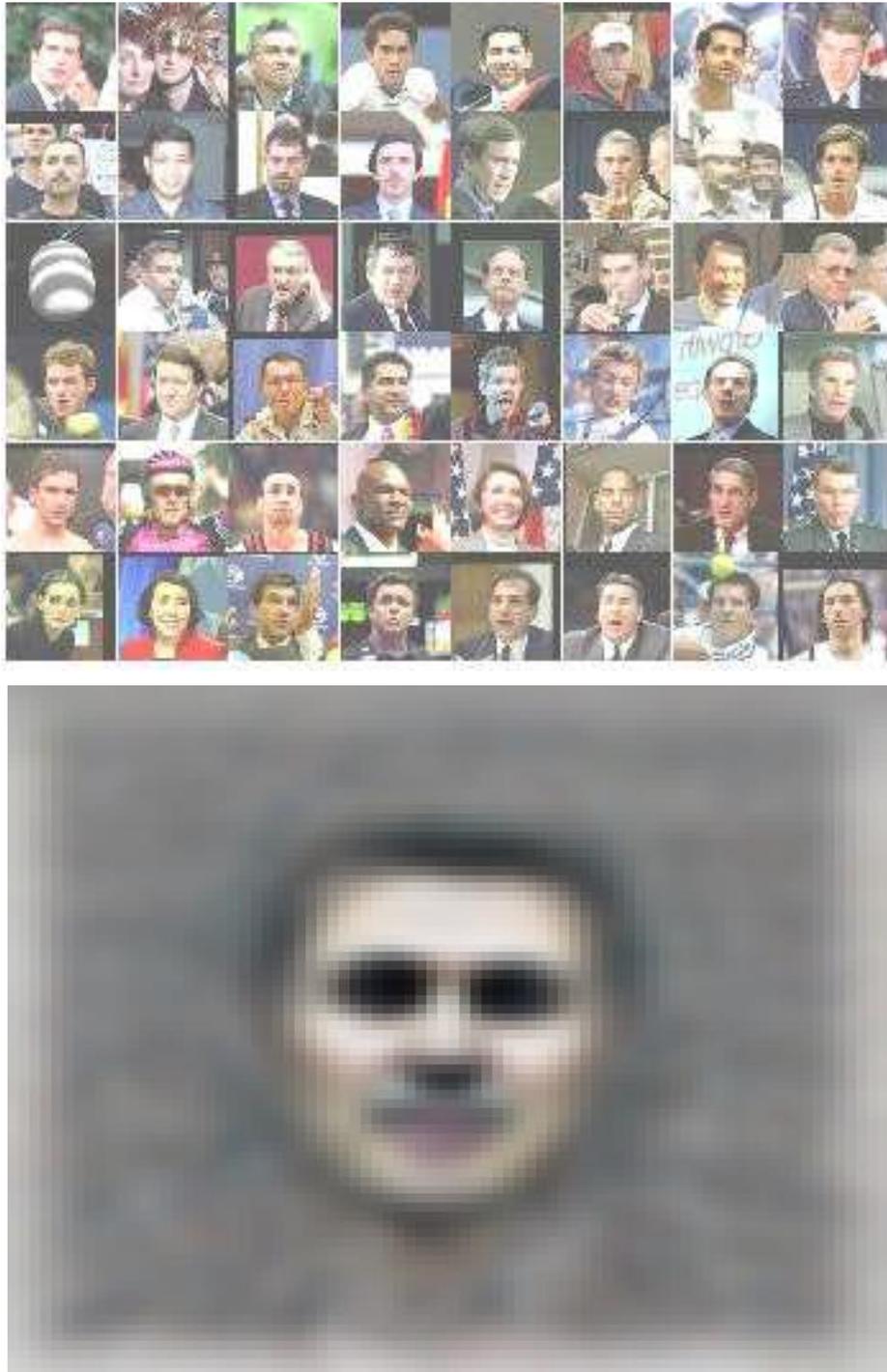


Fig. 8: 上部: ネットワークの一つのニューロンが最も強く反応する 48 の検証データ. その多くには顔が含まれており, このニューロンがヒトの顔に選択的であることがわかる. 下部: 顔に選択的なニューロンに最適な刺激の可視化データ. ヒトの顔のようなものが可視化されており, このニューロンが顔の概念を学習したことを意味する.

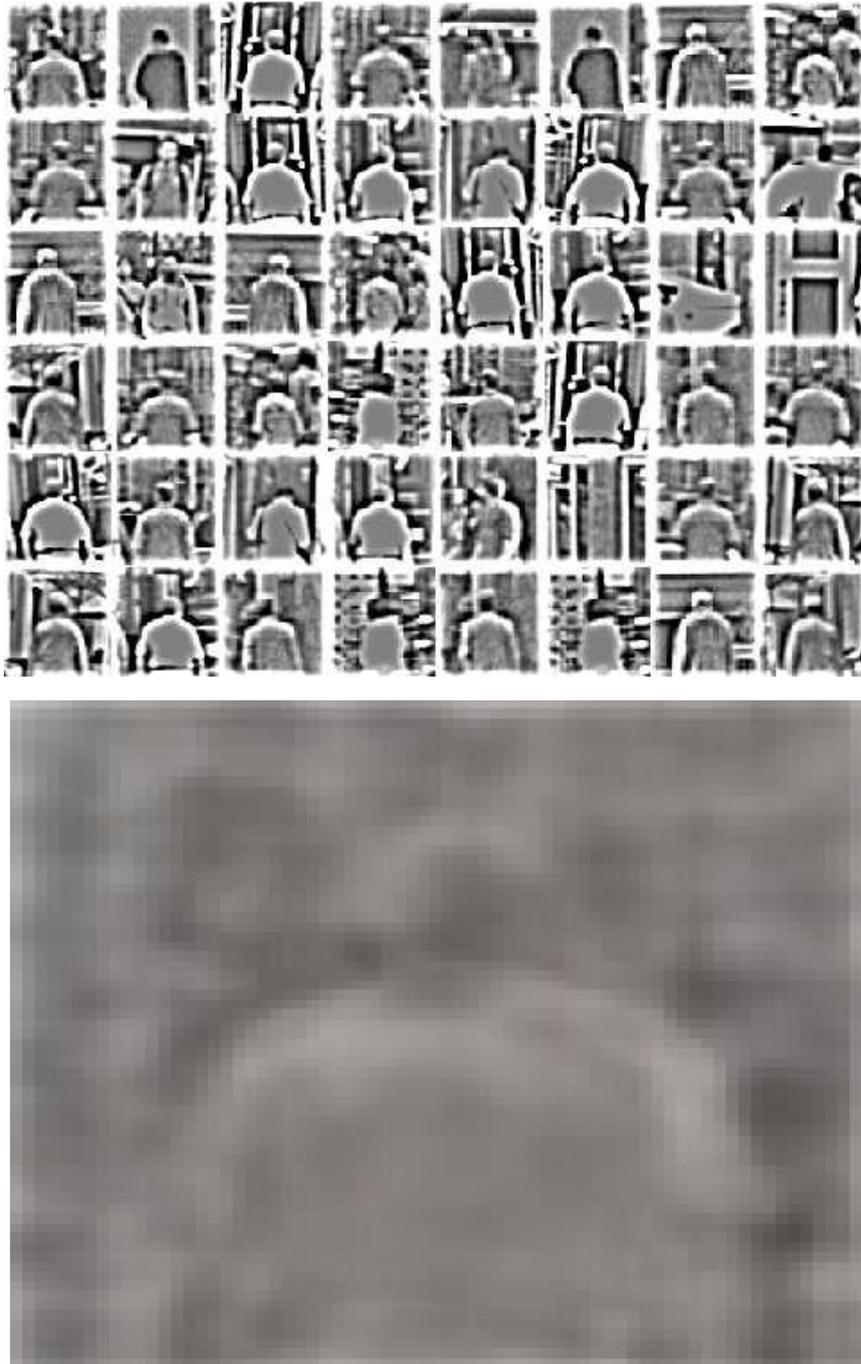


Fig. 9: 上部: ネットワークの一つのニューロンが最も強く反応する 48 の検証データ. その多くには人体のシルエットが含まれており, このニューロンが人体に選択的であることがわかる. 下部: 人体に選択的なニューロンに最適な刺激の可視化データ. ヒトの上半身のようなものが可視化されており, このニューロンが人体の概念を学習したことを意味する.

2.3.4. MT 細胞モデル

MT 野に存在するニューロンが持つ主要な特性の一つに速度選択性がある。MT の個々のニューロンはそれぞれ特定の方向に動く特定の速度の動きに強く反応する。このような MT ニューロンの特性を調べ、それらを再現するモデルがこれまで提案されてきた。その一つに Nishimoto らが提案した MT モデルがある [15]。ここでは、Nishimoto らのモデル構築手法について紹介する。

MT 野は視覚機能における運動視に関わる脳領野として知られており、これまで様々なモデルが提案されてきた。しかしこれまでの MT 野の研究では、実験に使用する動画やモデルの入力信号には、カメラで撮影したような自然な画像ではなく正弦波やその組み合わせである合成画像が使用されてきた。ヒトの自然環境下における動きに対する知覚特性を理解するには、より自然な画像を用いた実験とモデル構築が必要である。

Nishimoto らはこれを調査するために、より自然な画像を用いて実験を行った。

実験結果を踏まえて Nishimoto らはこの性質を再現するモデルの構築を行った。MT ニューロンを再現する最も単純なモデルは時空間ガボールフィルタリングとそれらの統合によって再現することができるが、上記のような複雑な刺激に対する応答を再現するには非線形関数をモデルに追加実装する必要がある。どのような非線形性が必要となるのか、これを理解するために筆者らはモデルに対して複数の非線形関数を自由に取り付け・取り外しを可能にするフレームワークを構築した。このフレームワークは基本的

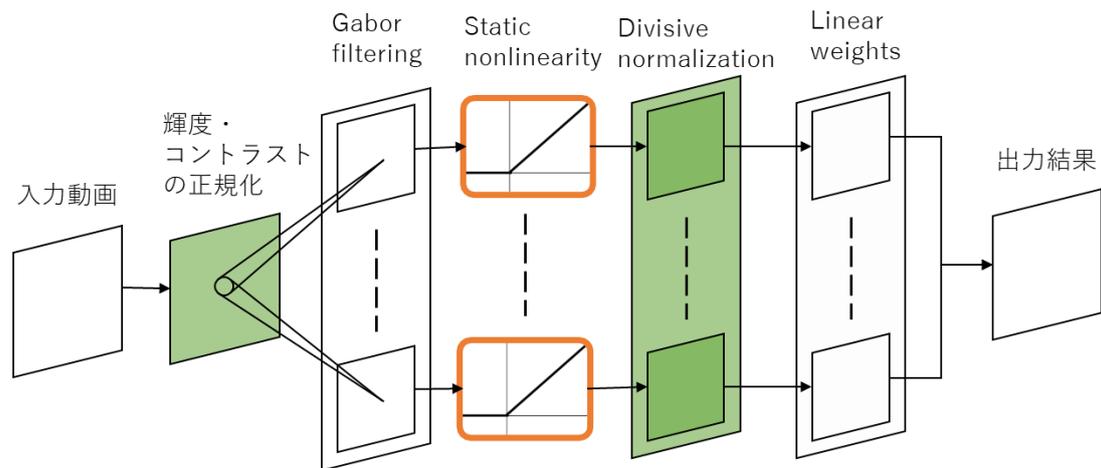


Fig. 10: フレームワークの構成図. 様々なガボールフィルタでフィルタリングする「Gabor filtering」、非線形変換の「輝度・コントラストの正規化」、「Static nonlinearity」、「Divisive normalization」、線形和を計算する「Linear weights」から構成されている。3種類の非線形変換の有無(緑の四角)または形状の違い(橙色の枠)によって、出力結果が実験結果をどれだけ正確に予測できているかを調べる。

にガボールフィルタと非線形変換、そしてそれらの線形和によって構成されている(Fig. 10). 非線形変換は「輝度とコントラストに対する正規化」、「除算型正規化」、「Static nonlinearity」の3種類あり、これらの違いやその有無による予測精度の変化を調べ、最も精度の高いモデルを明らかにする。

「輝度とコントラストに対する正規化」の数式を以下に示す：

$$I'(x, y, t) = \frac{I(x, y, t) - Lum(t)}{Con(t)}$$

$$Lum(t) = \sum_x \sum_y \frac{I(x, y, t)}{(X + Y)}$$
(6)

$$Con(t) = \sqrt{\sum_x \sum_y (I(x, y, t) - Lum(t))^2}$$

$Lum(t)$ と $Con(t)$ はそれぞれ輝度とコントラストを表し、入力 $I(x, y, t)$ に対して減算型と除算型の正規化を行っている。

「Static nonlinearity」は以下のように定義される：

$$X'(t) = \begin{cases} X(t)^\alpha & (t \geq 0) \\ 0 & (t < 0) \end{cases} \quad (7)$$

Static nonlinearity は半波整流であり、与えられた入力が0未満の場合は出力を0とし、0以上の場合は入力を α 乗したものを出力とする。ここでは α の値はそれぞれ、 $\alpha = 1.0$ (Linear), $\alpha = 2.0$ (Expansive nonlinearity), $\alpha = 0.5$ (Compressive nonlinearity)の三種類が用意されている。

「除算型正規化」の数式を以下のように表わされる：

$$X'(t) = \frac{X(t)}{\sum_n X_n(t) + \beta} \quad (8)$$

$\sum_n X_n(t)$ は除算型正規化の層に与えられるすべての入力の総和であり、個々の入力に対して除算を行うことで正規化する。

これら3種類の非線形変換の組み合わせを変えることで、実験から得られた結果の再現性の高いモデルを明らかにする。「輝度とコントラストに対する正規化」や「除算型正規化」の有無、「Static nonlinearity」の α の値の違いによって、12種類の組み合わせとなる(Fig. 11, Fig. 12, Fig. 13).

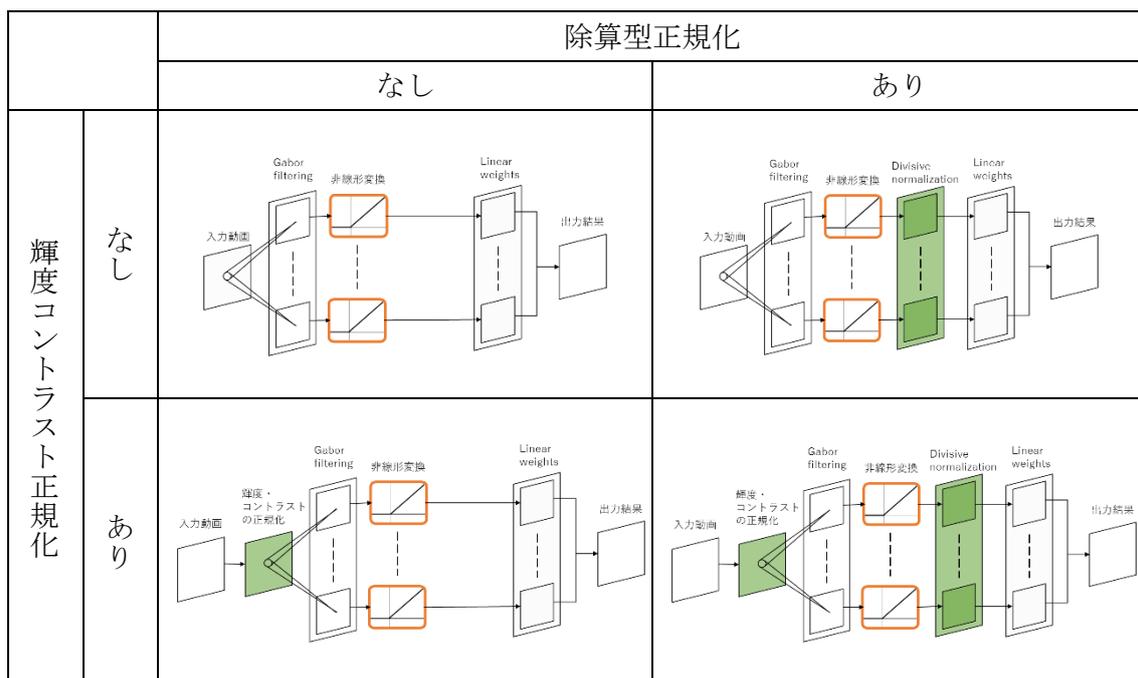


Fig. 11: $\alpha = 1.0$ の非線形演算を使用する場合のフレームワークの組み合わせ. 左上は輝度コントラスト正規化, 除算型正規化が共がない. 右上は除算型正規化のみ追加した. 左下は輝度コントラスト正規化のみ追加. 右下は輝度コントラスト正規化, 除算型正規化両方を追加.

12種類のモデルの予測精度結果について説明する。「輝度とコントラストに対する正規化」には有意な効果は得られなかった. しかし「Static nonlinearity」の違いでは有意な変化が見られた. Linear では有意な効果が見られなかったが, Compressive は予測精度を向上させ, Expansive では逆に減少している. また, 除算型正規化は性能を常に改善している. Compressive と除算型正規化の組み合わせは, より予測精度が改善される (Compressive 単体と比べて). これらの結果から「輝度とコントラストに対する正規化」なしの, Compressive な「Static nonlinearity」, 「除算型正規化」を含むものが最も再現性

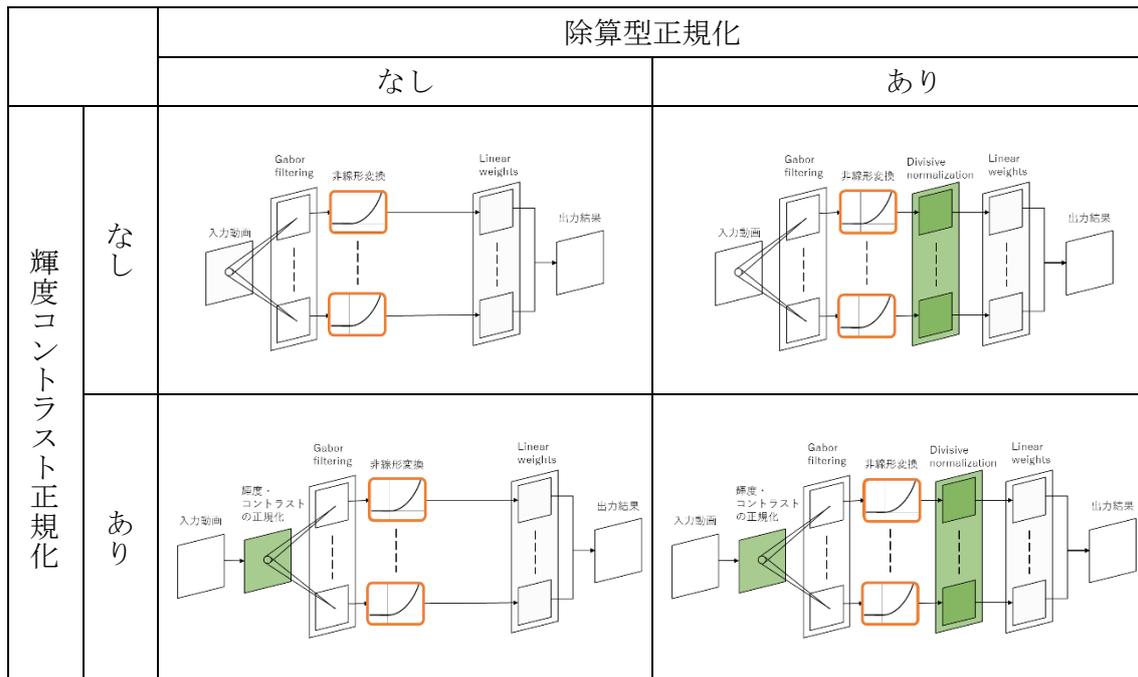


Fig. 12: $\alpha = 2.0$ の非線形演算を使用する場合のフレームワークの組み合わせ. それぞれの配置は Fig. 11 と同じ.

の高いモデルであることが分かった(Fig. 13)

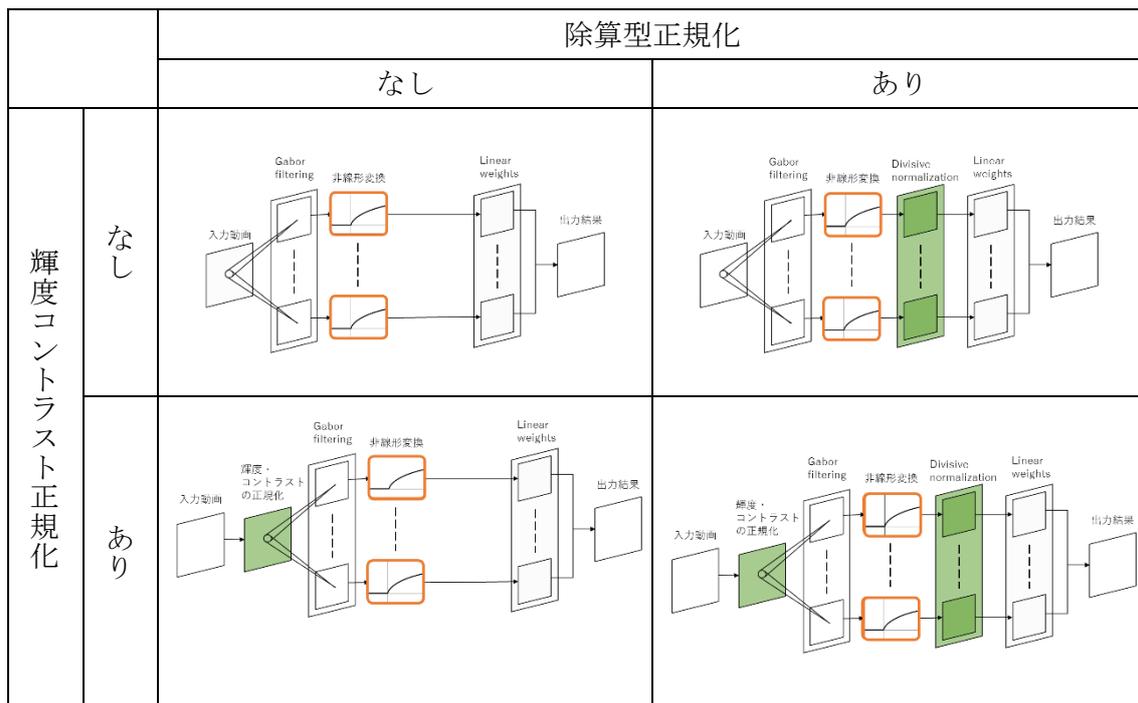


Fig. 13: $\alpha = 0.5$ な非線形演算を使用する場合のフレームワークの組み合わせ。それぞれの配置は Fig. 11 と同じ。右上の除算型正規化のみ加えたものは 12 種類の組み合わせの中で最も予測精度が高い。

2.4. まとめ

視覚に関わる疎粒度モデルをいくつか紹介してきた。Fig. 2, Fig. 5, Fig. 7, Fig. 10 のネットワークモデルの構造に共通することは、「複数の構成要素(コンポーネント)を階層的に組み合わせて構築している」ことである。例えば、Neocognitron は「S 細胞層」と「C 細胞層」の 2 つの要素で構成され、Saliency マップは各特徴に対する「スケール間での相関計算」, 「空間的競合」, 全特徴の「線形和」の 3 要素で作成される。Le らのディープニューラルネットワークは「局所受容野」, 「プーリング」, 「局所コントラスト正

規化」の繰り返しであり、Nishimoto らの運動選択性モデルでは、構成要素として「輝度・コントラスト正規化」、「非線形演算」、「除算型正規化」を挙げ、これらの組み合わせからなるモデルをすべて評価することで電気生理実験結果を最もよく再現するモデルを選択している。

以上のようにモデルのコンポーネント性は多くの数理モデル構築に共通している。コンポーネントの組み合わせによるモデル構築は前述のように、工学的モデルとしても神経生理的モデルとしても有効であり、手段としても有用である。その他の新規数理モデルを構築する場合でも、モデル全体を明示的にコンポーネントとして分割することで、作成や修正が容易になる。一つのモデルを一つのプログラムとして記述したモデルより、複数のコンポーネントとその結合によって構成されたモデルのほうがモデルの構造やデータの流れが理解しやすいため、他者がそれを再利用する場合にも有益である。また複数のコンポーネントで構成されたモデルをまた別のモデルのコンポーネントとすることで、モデルの拡張が容易となる。これらの操作を繰り返すことで、より大規模なモデルの構築が可能となるだろう。そこで本研究では、モデルのコンポーネント性に着目し、コンポーネントの組み合わせによるモデル構築を可能にするソフトウェア基盤を構築する。

3. HI-brain の開発方針

前章で明らかになった視覚数理モデルの「コンポーネント性」、ならびに「コンポーネント化による利点」から、モデル開発にはコンポーネントの組み合わせによって構築する環境が適していると考えられる。本章では、ソフトウェア基盤の必要要件と機能について調査する。

3.1. プラットフォームの開発方針と要件

モデルのコンポーネント性は既存疎粒度モデルの共通点である。このことから、モデルを複数のコンポーネントの組み合わせによって構成する「コンポーネント指向開発」が有益であると考えられる。しかし現状の疎粒度モデルの開発環境ではそれを行うことは難しい。疎粒度モデルの共通ソフトウェア基盤は存在しておらず、モデルの開発言語や入出力データフォーマット、パラメータの記述形式などのモデル規格は統一されていない。故にモデル間の結合は難しく、その動作環境も研究者ごとに異なる。そのため現在公開されているモデルの実行プログラムを結合する前に、それを自らの環境で動作させなければならないがそれは非常に困難である。

本研究で構築するソフトウェア基盤は、個々のコンポーネントの結合・置換によってモデル構築を行うことができ、作成したモデル同士の結合によってより大規模なモデル構築を可能とする。具体的には、Fig. 1-A~C のようにモデルの結合とそのシミュレーションが可能であり、また Fig. 1-C の MT と newMT のようにモデルの置換も同様に行え

る環境を指す。まずは必要となる要件をまとめ、それを実現するために必要な要件または機能を上げる。

ソフトウェア基盤の必要要件として、「結合・置換によるモデル構築」、「既存モデルの再利用」による大規模化、構築したモデルの「大規模シミュレーション」があげられる。また必要要件ではないが、その基盤の利便性についても考慮する必要がある。以下に必要要件とそれを満たすための機能について記載する。

I. モデルの結合・置換

研究者は既存モデルをシミュレーションするために、モデルをコンピュータプログラムとして実装する。これらモデルを結合させるには単純に一つのプログラムコードとして記述しなおす必要がある。結合させるモデルの数が膨大になると、その手間もまた膨大となる。各研究者が作成するモデルの結合・置換を可能とする互換性があれば、結合に必要な手間を減らすことができる。そのため、以下の要件を必要とする。

I-I モデルの標準化

国際標準化団体などによりモデル記述やデータ記述の仕様が規格化されていること。視覚モデルを一つのコンポーネントとして使用できるように、また各研究者による共同開発を容易にするためには、各モデルの仕様をあらかじめ統一しておく必要がある。モデルの入出力データやパラメータ、それらデータの格納先

の指定方法が規格化されており、シミュレーション時の状態遷移(初期化, 実行時処理, 実行終了時処理など)やそこで実行されるプログラム関数名が規格化されていること。

脳科学研究が今後永続的に行われることと同様に, 数理モデルも今後各研究者によって提案されることが予想される。しかし, 脳研究者の開発環境は統一されていない。個々のモデルに互換性を与えるには, 開発言語またはデータフォーマットの国際標準仕様の策定が必要である。しかしこれらモデル形式の統一をするには, 膨大な事前調査や合同会議, ドキュメントの整備, 広報活動など多大な労力が必要となる。しかし, それがなくとも小規模な(微に入り細を穿つような)モデル研究は可能であるため, これまでにそのような動きは確認されていない。そこで本研究では既存の国際標準仕様を従った“脳研究に限定しない”ソフトウェア基盤を拡張し, 脳数理モデルに応用する。

モデルの互換性のためには, 標準仕様に従ったモデル開発が必要であるが, それだけではモデル間の結合は難しい。モデルの入出力データの形式を共通させる必要がある。使用する標準仕様は脳研究のみを目的としているのではないため, 本研究が対象とする疎粒度モデル用のデータ型を定義しなければならない。よって次の機能が必要である。

I-II 疎粒度モデルのデータ記述

個々のモデルが転送するデータはモデルごとに異なり, それらは大きく2種類に分かれる。一つは画像配列データである。これには画像のメタデータと同様に

色空間やビット深度などを含む。もう一つは多次元配列データである。これはモデルの入出力や内部状態データを記述する上で有用である。また整数や実数などの配列要素の型の情報も含まれる。これらすべてを記述できるデータ型を定義し、モデルのコンポーネント性を高める。

モデルの標準仕様及びデータ型の定義によって、モデル結合が可能となる。しかし新たなデータ型を定義する場合、モデル形式の統一と同様に多大な労力を必要とする。そこで、既存のデータ型を基にソフトウェア基盤で使用するデータ型を定義する。

そして結合させたモデルを一つのモデルとして正確にシミュレーションしたいと考えるだろう。また結合させるモデルが増えることによるモデルの大規模化も考慮する必要がある。

II. モデルの大規模化とシミュレーション

結合するモデルが増加することによる弊害への配慮と、結合させたモデルの正確なシミュレーションを可能とする。現在の疎粒度モデル研究では個々の脳機能を対象としている。しかし将来的には脳全体のモデル化とシミュレーションを行うと考えられる。構築する基盤でそれが不可能な場合、また新たな基盤の構築と既存モデルの再実装をする必要がある。本研究で構築するソフトウェア基盤は疎粒度モデルを対象とするが、他の感覚野(聴覚や運動制御など)との結合も可能とする。

モデルのシミュレーションには、正確にシミュレーション可能であることと現実的な時間内に実行可能であることが重要である。モデルのアルゴリズムを記述したプログラムを実行することが一般的なシミュレーションであるため、シミュレーションの正確性について考慮する必要はない。しかし本基盤では、結合させたモデルを一つのモデルとしてシミュレーションするため、正確性を考慮する必要がある。またシミュレーション対象の大規模化に伴い、計算時間の増加が懸念される。そのため、現実的なシミュレーション時間で行うための機能が必要となる。

II-I 時間同期シミュレーション

結合関係によって構成される数理モデルに対し、シミュレーションの再現性を保障する。Fig. 14-A のような結合関係の場合、各コンポーネントが定期的に行われるのではなく (Fig. 14-B) , 入力データが与えられるのを待ってから実行するようにしたい (Fig. 14-C) 。そのためコンポーネントの実行周期を管理する機能が必要とされる。

II-II 汎用性

モデルの大規模化による聴覚情報処理や運動制御などの数理モデルも同様に、結合・置換・シミュレーションが可能であること。そのためにモデル間で授受されるデータ型を視覚用、聴覚用、運動制御用など目的に応じて定義可能であるこ

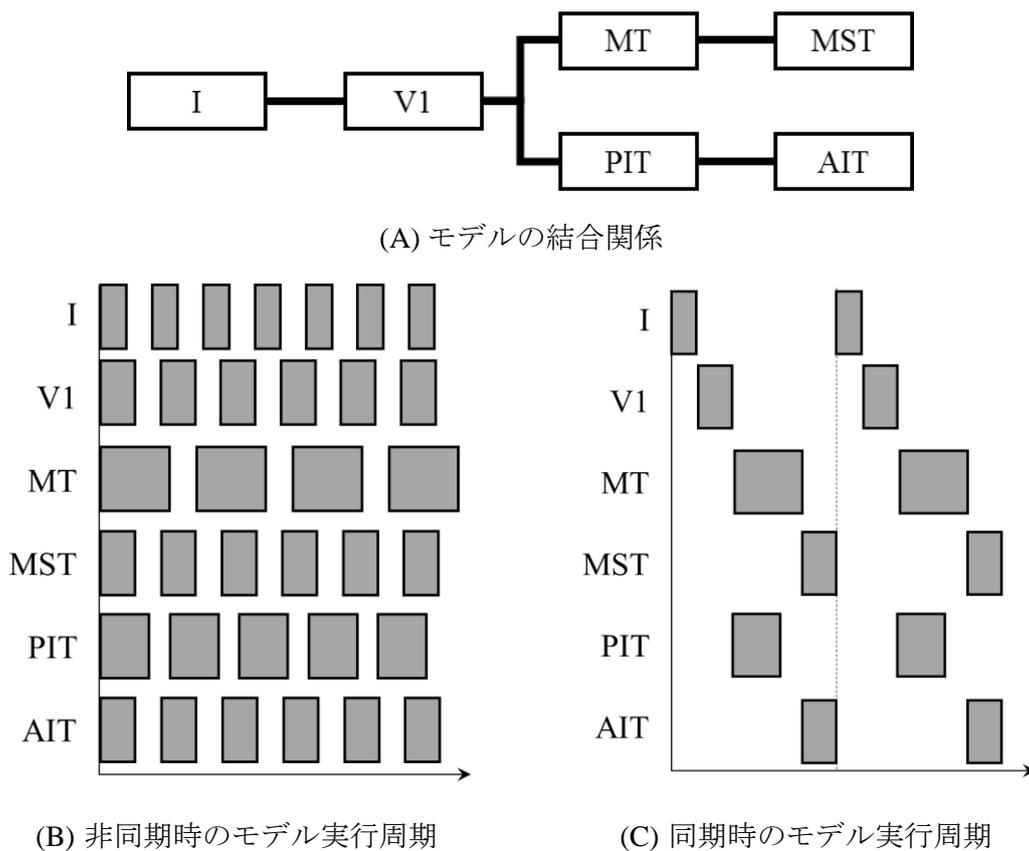


Fig. 14: モデルの結合関係と同期・非同期時のモデル実行周期. (B), (C)の横軸は時間を意味し、バーの長さがそのコンポーネントの1周期当たりのシミュレーションに要する時間を表す. (A)の結合関係にあるモデルを正確にシミュレーションする場合、(B)のような独立した実行周期ではなく、(C)のように結合状態に応じてモデルの実行周期を変える必要がある.

と.

II-III 高速シミュレーション

シミュレーション対象の拡大によって結合されるモデル数が増加する. これによる実行時間の増加を軽減するための、並列分散処理やモデル間データ通信の高速化が可能であること.

III. 既存モデルの再利用

複数のコンポーネントを組み合わせた新規モデルを構築する場合、すべてのコンポーネントを1から構築するのではなく、これまで作成されてきたモデルやモデル作成にライブラリ関数を再利用したいと考えるだろう。MATLAB, C/C++, Python などの異なるプログラミング言語で記述された既存モデルを再利用するための手法を必要とする。

III-I 資産継承

モデルの作成に既存のライブラリ関数を使用するには、各関数を結合・置換可能であるようにコンポーネント化する必要がある。これには手間がかかるため必要な既存画像処理ライブラリを容易に利用することができる機能が必要となる。

III-II 研究者間でのモデル共有

研究者間でのモデル共有が可能であること。本基盤を使用したモデル開発によって、各研究者が作成したモデル間の結合が可能である。そのためモデルを共有させるためのデータベースが必要となる。

これらの要件をみたす、もしくは要件をみたすことが期待される既存プラットフォームを見出すために調査、考察した結果を次節に記す。また必要要件ではないが、本基盤の使いやすさや導入の手軽さなどのソフトウェア基盤の利便性もまた考慮する必要がある。

IV. 利便性

基盤導入時やモデル作成時に複雑な手順を必要としないこと、対象が脳研究者であることを考えると、基盤導入時やモデル作成時に複雑な手順を必要としないものが望ましい。そのため、環境構築やモデル作成時の手間を最小限にする工夫が必要である。

IV-I 視覚モデル用データフォーマットの相互変換

視覚モデル構築で主に使用されているデータフォーマットの相互変換が容易であること。具体的には、視覚モデル開発のデファクトスタンダードである MATLAB や OpenCV で使用されているデータフォーマットを、要件 I-II で作成するデータ型へ変換するためのライブラリを作成する。

IV-II 必要最低限の手順での基盤導入方法

必要最低限の手順でソフトウェア基盤の導入を可能とすること。環境導入用のインストーラや OS などのソフトウェアが用意されていること。

IV-III 標準規格に従ったモデル実装の工程削減

要件 I-I の標準規格に従ったモデル作成時に必要な手順を省くこと。モデルの結合・置換を可能にするには標準規格の導入が重要である。また要件 III-I を満たすためには、既存のライブラリやモデルに対して標準化を施す必要がある。しかしそのための工程が複雑であることは避けるべきである。そこでその工程を自動化

し実装作業を単純化するツールを作成する。このツールは多種多様な疎粒度モデルを対象とする。

3.2. 既存ソフトウェア基盤と必要要件

「モデルの結合・置換」が可能な既存プラットフォームについて調査を行い、必要となる機能についての比較検討を行った。Table 1 内の○印は要件を直接満たしていることを示しており、●印は要件を直接満たしていないが、新規開発により要件を満たす拡張性があることを示している。以下に Table 1 の主要点を記す。

既存の視覚モデルは MATLAB で記述されていることが多い。そのためモデル結合には結合・置換を可能にする Simulink が有効であると考えられる。そのためには標準化を行う必要があるが、前述したように現実的ではない。

NEURON⁸や GENESIS⁹は神経細胞シミュレータであり、「疎粒度モデル」ではなく「細粒度モデル」を構築の対象としている。しかし、その拡張性次第では疎粒度モデルの構築も可能ではないかと考え、調査の対象とした。しかし、独自の記述言語やデータ型を採用しているため標準化には該当せず、資産継承、汎用性の観点でも不足していることが分かった。PLATO[42]は本研究と同様の視覚モデル用開発基盤である。しかし、必要機能 I-I, II-III および III-I を満たしていない。

⁸ <http://www.neuron.yale.edu/neuron/>

⁹ <http://genesis-sim.org/>

必須要件を満たすための機能すべてを満たす既存視覚モデル構築基盤は存在しなかった。しかし、OpenRTM-aist は必要機能を新規開発することによって満たすことができる。事前調査によると、OpenRTM-aist では、視覚モデル用データ型を IDL(Interface Definition Language)を定義することで機能 I-II を実現することができる。同様に、実行

Table 1: 必要要件と既存基盤。○はその基盤が要件を満たしていることを意味し、●はそれを満たしうる拡張性があることを表す。

		MATLAB / Simulink	Neuron	GENESIS	PLATO	OpenRTM-aist
モデルの結合・置換	モデルの標準化					○
	疎粒度モデルのデータ記述	●	●	●	○	●
モデルの大規模化とシミュレーション	時間同期シミュレーション		○	○	○	●
	汎用性	○			○	○
	高速シミュレーション	○	○	○		●
既存モデルの再利用	資産継承	●				●
	研究者間でのモデル共有	●	○	○	●	●

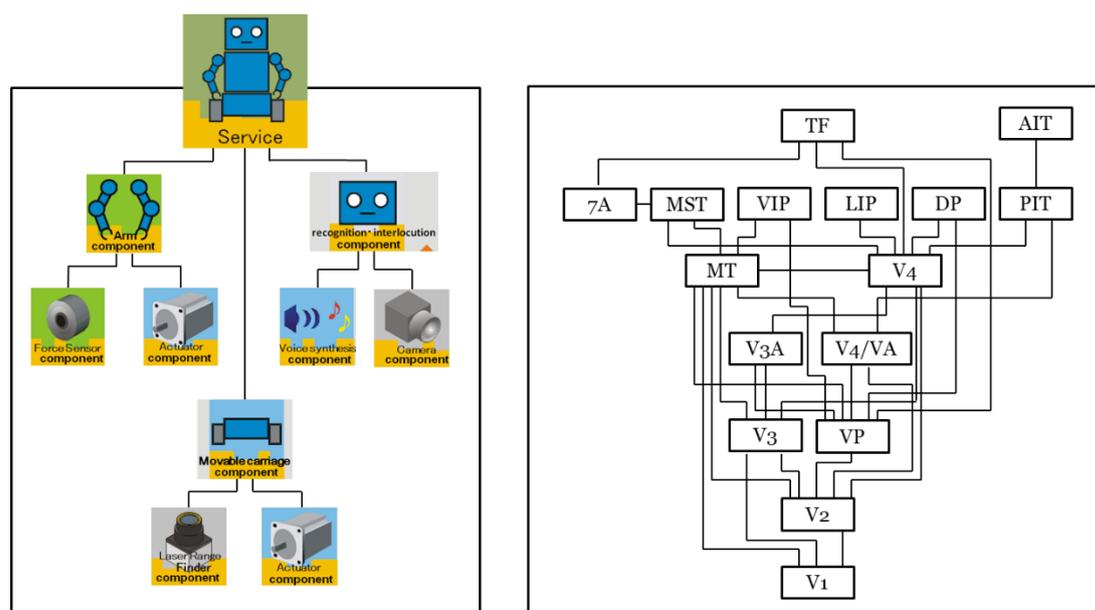


Fig. 15: ロボットシステム構築と大規模視覚モデル構築の類似点. 複数のパーツの協調的動作によって一台のロボットが動作する. 同様に複数のモデルの協調的に動作によって視覚システムのような複雑なモデルを構築することが可能であると考えられる.

コンテキストと呼ばれる OpenRTM-aist の機能を用いれば機能 II-I が実現できることを見出した.

本研究では RT ミドルウェアと呼ばれる規格, ならびにその実装の一つである OpenRTM-aist を基盤とし, 視覚数理モデル構築に必要な機能を追加することとした. 次節で具体的に RT ミドルウェア・OpenRTM-aist と視覚数理モデルの親和性を記す.

3.3. RT ミドルウェアと視覚数理モデルの親和性

OpenRTM-aist は RT ミドルウェアと呼ばれる, ロボットシステム開発用ミドルウェア規格に従っている. ロボット開発では, 脳やその機能である視覚と同様にセンサーやア

クチュエータなどのコンポーネントとそれらの結合関係によって構成されている。脳を構成する各モデルを構成要素として結合させ、複雑な機能を持つ脳のシミュレーションが可能ではないかと考えた(Fig. 15)。RT ミドルウェアとその実装である OpenRTM-aist が本研究の目的を達成する基盤であることが期待できる。コンポーネントを記述する言語も C++, Python, Java の多言語に対応しており、公開されている既存モデルのように様々な言語で記述することができる。OpenRTM-aist はオープンソースで公開されており、さらなる機能の拡張が見込まれる。

OpenRTM-aist でのシステム開発では Fig. 16 のように GUI のエディターが用意されている。エディター上ではコンポーネントの結合や実行を行うことができる。四角がコンポーネントを表しており、結合しているコンポーネント同士が線でつながっている。

OpenRTM-aist は（視覚モデル用基盤ではないのだが）重要な要件である「モデルの結合・置換」が可能であることが分かった。このコンポーネントとインターフェースは OMG によって標準化され、公開されている [43,44]。これで作成されたモデル及びコンポーネントは国際標準規格を満たしていることになるため、要件 I-I を満たすことになる。

OpenRTM-aist の目的はロボットシステム開発である。実際のロボットの実績は十分であり、ヒトの運動制御や他感覚情報処理モデルの実装なども十分可能であると考えられる。よって要件 II-II を満たしているといえるだろう。視覚数理モデル開発が目的

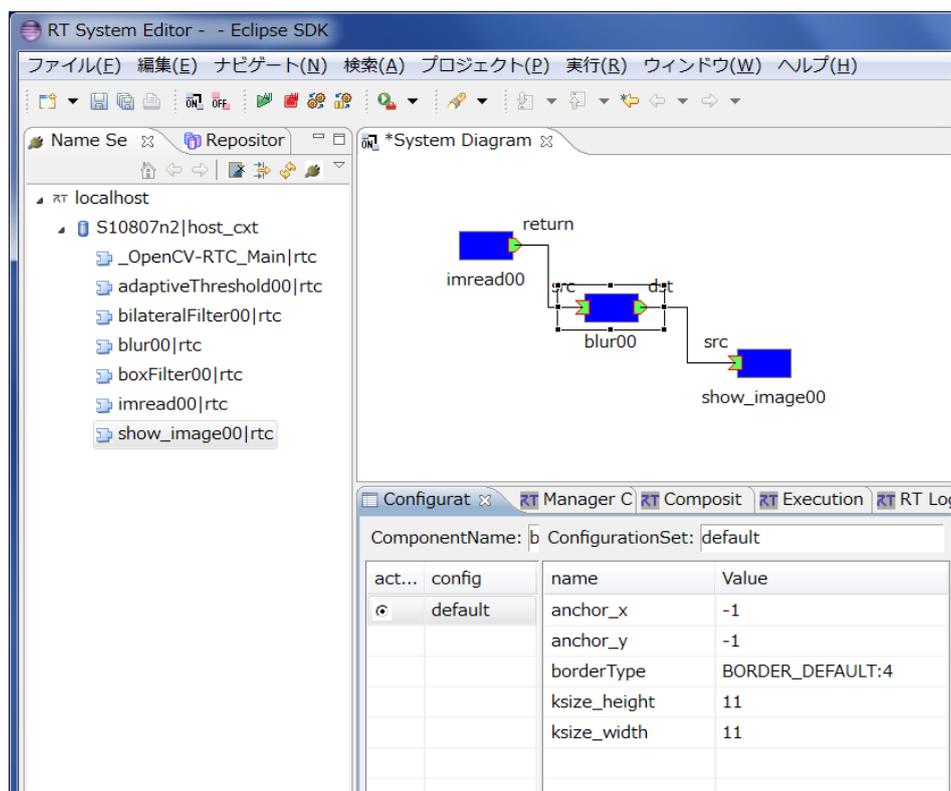


Fig. 16: OpenRTM-aist の GUI エディターが表示するコンポーネント結合の様子。左側のビューに現在起動しているコンポーネントの一覧が表示されてる。右上の青い四角と線はコンポーネントとその結合を表す。ここでは画像読み込み用のコンポーネントからぼかし処理をするコンポーネント、画像表示用のコンポーネントへ結合されている。その下のビューでは選択したコンポーネントの変数が表示されている。

ではないデメリットとして、本研究の目的を達成するための機能をいくつか追加開発する必要がある。次節以降では我々が開発した新たな機能と結果を記す。

4. OpenRTM-aist の機能拡張

前章では、OpenRTM-aist が本研究の目的を果たすためには最適であると判断した。しかし脳科学研究での実績がなく、それを想定した機能構築がされていない。Table1 に●で示した要件を満たすために新規開発や定義を行った。

4.1. 視覚モデル用の共通入出力データ型

要件 I-II を達成するための調査を行った。OpenRTM-aist ではデータの送受信に使用するデータの型をそれぞれのユーザが自由に定義することができる。OpenRTM-aist には標準で用意されているデータ型がいくつか存在する。その一つである `CameraImage` 形は標準で用意されている画像転送用のデータ型である。しかし `CameraImage` 型には配列要素の型情報を格納することができない。つまり、結合自体は可能だが、送られてきたデータが整数なのか実数なのかなどの判断材料がないため、要件を満たすことができない。これが原因で本来求められている処理とは異なる結果を出力するもしくはエラーが発生することが考えられる。

要件 I-II を満たす既存データ型の候補として、次のフォーマットの比較検討を行った。

- (A) MATLAB 言語の配列フォーマット（以降、便宜上 `MATLAB::mat` と記す）。
- (B) 画像処理ライブラリ `OpenCV` の配列フォーマット（以降 `cv::Mat` と記す）。

(A)は多次元の行列を格納するためのデータ型である。しかし画像配列データを格納

する際に必要となる色深度や色空間の定義は存在しない。(B)の `cv::Mat` は画像メタデータと画像自体のデータを格納することができる。さらに配列要素の型が明確に定義されているため、要件 I-II を満たすと判断できる。例えば `CV_8U` は符号なし 8bit 整数型、`CV_32F` は単精度浮動小数点数型などと定義されている。多次元配列データの格納についても、入出力データ及び内部状態データを `cv::Mat` の配列に格納することができる。

これらの理由から、疎粒度モデルの開発で広く使用されている MATLAB の配列フォーマットではなく、画像処理ライブラリである OpenCV の配列フォーマットである `cv::Mat` を疎粒度モデルの共通配列データ型として採用した。OpenRTM-aist のデータ型として `cv::Mat` を使用するために、`Timed_cvMat` 型を定義した(Table 2)。この `Timed_cvMat` 型には、データ通信の方式を格納するための変数を含んでいる。これは要件 II-III を満たすためであり、4.3 節で説明する。

また `Timed_cvMat` 型は要件 III-I を満たすためにも重要である。OpenCV ライブラリには、モデル研究において構成要素となる相関計算や画像フィルタなどの各種画像処理関数や網膜細胞モデル[10]が用意されている。OpenCV で一般的に使用される配列フォーマットをデータ型とすることにより、OpenCV 関数のコンポーネントの入出力データを複雑な変換なしに授受させることができる。2016 年現在でも開発が行われており、OpenCV で利用できるアルゴリズムやその関数なども増加している。ゆえに `Timed_cvMat` 型は要件 III-I を満たすために必要となる。

4.2. 複数モデルの同期実行機能

結合させたモデルのシミュレーションの正確性を維持するには、複数のモデル出力が合流する結合をさせた場合にモデルの同期性の保証が重要である。同期性がない場合は、科学研究において重要な要件である再現性が保証されない。OpenRTM-aist にはコンポーネントの実行周期や状態遷移を管理する「実行コンテキスト」というものが用意されている。既存の実行コンテキストとして、Periodic ExecutionContext(Periodic EC) が存在しているが、これは結合されたモデルを正確にシミュレーションすることができない [45](Fig. 17).

Table 2: 視覚モデル用入出力データ型である Timed_cvMat 型の構成。OpenCV の標準多次元配列クラスである cv::Mat を含む。

型名	変数名	説明
Timed_cvMat		
Time	tm	タイムスタンプ
cvMat	data	
string	data_type	データ型名(C++のtypeidより取得)
long	mat_type	Matのデータ型
short	width	Matのサイズ(幅)
short	height	Matのサイズ(高さ)
short	bpp	Matのサイズ(1ピクセルあたりのバイト数)
boolean	is_shared	共有メモリ使用有無
		trueの場合、共有メモリを使用する
		falseの場合、共有メモリを使用しない
string	name	共有メモリ領域名
		(is_shared=falseのときは使用しない)
long	memory_index	共有メモリのリングバッファインデックス
		(is_shared=falseのときは使用しない)
long	length	Matデータ長
sequence<octet>	data	Matデータ
		※OpenCV関数の入出力がVector型の場合も、Mat型に変換してポート間の入出力を実施する。
		(is_shared=falseのときは使用しない)

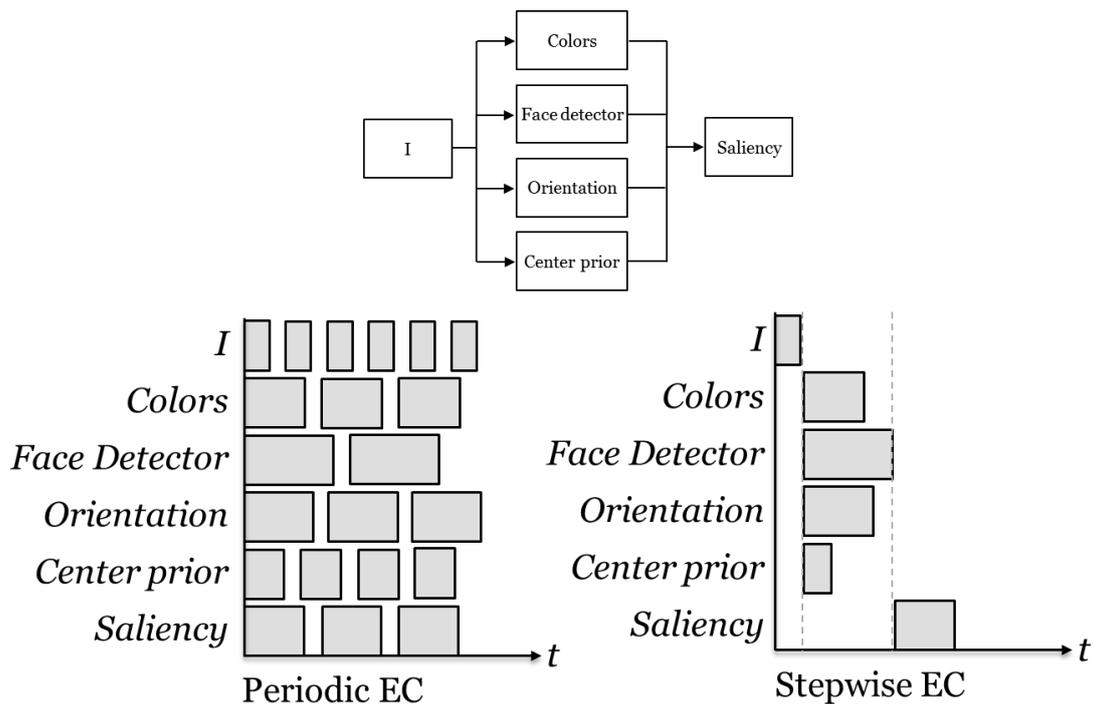


Fig. 17: 実行コンテキストごとのモデル実行周期の比較. Periodic Execution Context (Periodic EC)の場合, 各モデルが独立してモデルを実行させるのに対して, Stepwise Execution Context (Stepwise EC)の場合は, 各モデルが新たな入力データを取得した後にモデルを実行させる. Stepwise EC を使用することでモデルの周期実行が可能となる.

そこで本研究ではモデルの同期生の確保, シミュレーション時の一時停止やコマ送り実行 (1 ステップ実行) を可能とする実行コンテキスト, Stepwise Execution Context(Stepwise EC)を開発した (Fig. 17). OpenRTM-aist では, コンポーネントの状態管理を制御するために実行コンテキストが使用される. これは OpenRTM-aist の機能の一つであり, 開発者が目的に応じて設計することができる. 既存の実行コンテキストである Periodic EC と本研究で開発した Stepwise EC では, 各コンポーネントに対して1つ

の実行コンテキストが管理する。しかし Periodic EC では各実行コンテキストがそれぞれのタイミングで状態遷移命令を出す。そのため同期性が保証されず、各タイミングで与えられた入力すべてに対して処理を行うことができないなどの問題が発生する (Fig. 15 右)。Stepwise EC では実行コンテキスト(master)が各コンポーネントに割り当てられている実行コンテキスト(slave)を管理する。master が状態遷移命令を slave に対し適宜出すことで、結合関係全体の処理が同期される。

OpenRTM-aist の標準機能でもモデルの同期性の確保が可能である、一つは「複合コンポーネント機能」を使用する手法である。これは複数のコンポーネントを一つのコンポーネントとして扱い、それらコンポーネントを同期処理させる。しかし、20 個程度以上のモデルを複合コンポーネントで一つにまとめ、同期実行させた場合、実行に要する時間が長い。またフィードバック結合を行った場合、デッドロックが発生する可能性があり、それを回避するための初期値設定を柔軟に行うことが困難である。

もう一つの方法として ExtTrig ExecutionContext を使用する方法がある。これは外部トリガを使用して、コンポーネントの実行を制御することができる。しかし、外部トリガを扱うためのプログラムが別途必要になる。これはコンポーネントの動作を把握しないため、同期性の確保は難しい。

4.3. データ通信の高速化

疎粒度の視覚モデルは、入出力データに画像や動画などを使用する。これらは画像サ

イズや動画の長さによってデータ量が膨大となり、データ転送に必要な時間も同様に膨大となる。またモデルの大規模化に従ってモデル間の結合が増加するため、データ通信速度の影響が顕著になる。そこでデータ転送に必要な処理時間を短縮する必要がある。また、多数のモデルを結合させた大規模シミュレーションの実行時には、計算機への負荷の増加も問題となる。解決のためには、複数の計算機に負荷を分散させる並列分散処理による計算時間の短縮が必要となる。

OpenRTM-aist は規定の通信方式として CORBA を採用することで並列分散処理を可能としている。CORBA は異なる言語及び OS で動作するプログラム間のデータ処理を可能にする国際標準アーキテクチャである。これはソフトウェアの国際標準化団体 OMG(Object Management Group)によって標準規格として採択されている。また CORBA では IDL(Interface definition Language)を利用して入出力データフォーマットを自由に定義することができる。これもまた OMG によって採択されている。CORBA は共通のプロトコルを使用しており、CORBA に従ったモデル間のインターオペラビリティを実現している。これを用いることで多数のモデルを結合させた大規模シミュレーションを行う際に、ネットワーク結合させた計算機を複数使った並列分散処理による計算の負荷の分散と時間の短縮を可能にする。

OpenRTM-aist で用意されている通信方式は複数の計算機を使用した並列分散処理を行う場合にメリットがある。しかし単一の計算機を使用したシミュレーションの場合で

も、同様に分散メモリを使用し、計算時間が増加する。そこで我々は、共有メモリによるデータ送受信が可能であればそれを使用し、可能でなければ CORBA 通信を使用するソフトウェアライブラリを開発した。共有メモリに関する実装方法は Windows API や Qt¹⁰など様々あるが、移植性の観点から boost¹¹と呼ばれるライブラリを利用して実装した。

具体的な開発内容について記述する。コンポーネントの入出力データポートクラスを継承することで、新たなデータポートクラスを作成した。このデータポートクラスを利用することで、boost の共有メモリ機能を利用したデータ通信が可能となる。このクラス内にはコールバック関数を新たに追加している。この関数はコンポーネントの接続時または切断時に呼び出される。その際にコンポーネント間の通信方式が自動判別される。例えば、結合させる 2 つのコンポーネントが同一の計算機に存在するのであれば共有メモリ、異なる計算機上でのデータ通信であるならば OpenRTM-aist の従来の通信方式 (CORBA 通信) を使用させる。データポートクラスを新規に開発するのではなく、既存のクラスを継承し実装することで、必要最低限の手間でかつユーザが規定の使用方法で使用することができる。

実際にこの機能を利用してデータを転送した場合と既存の手法を使用した場合では、

¹⁰ <http://www.qt.io/>

¹¹ <http://www.boost.org/>

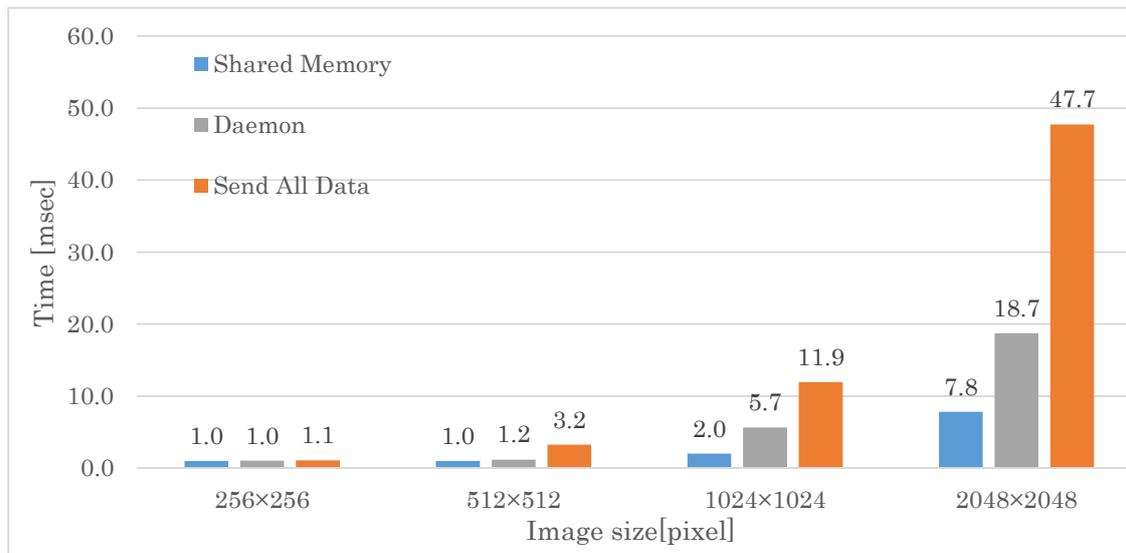


Fig. 18: 転送方式と転送時間の関係. グラフの青は共有メモリを使用した場合, 灰は RTC Daemon を利用した場合, 白は OpenRTM-aist で標準転送方式を使用した場合の転送時間をそれぞれ表している.

データの転送速度にどのように影響するか, それを調べるために転送に要した時間を測定した. Fig. 18 にそれぞれの方法を使用した場合のデータ転送時間を示す. 転送するデータの型は 4.1 節に記述した `Timed_cvMat` 型を使用した. 転送時間は, 送信側コンポーネントの `cv::Mat` データが転送可能となった時点から計測を開始, 受信側コンポーネントが転送されたデータを `cv::Mat` に変換した時点を終了とする. すなわち, `cv::Mat` から `Timed_cvMat` 型への変換, 共有メモリへの書き込み, 転送時間, 共有メモリの読み込み, `Timed_cvMat` 型から `cv::Mat` への変換を行う合計時間を計測する.

測定結果は共有メモリを使用したデータ通信方式で転送時間を短縮することを肯定する. 例えば, 1024x1024pixel の静止画像 (3072 KiB ; 色チャンネル数 : 3, 色深度 : 8bit)

をそれぞれの通信方式で転送した場合、データを全転送する場合の転送時間は 11.9ms、RTC Daemon を使用した場合の転送時間は 5.7ms であったのに対し、共有メモリを利用した通信方式での転送時間は 2.0ms であった。同様に画像サイズが 2048×2048 pixel である場合は、データの全転送で 47.7ms、RTC Daemon で 18.7ms、共有メモリでは 7.8ms の結果となり、画像サイズが大きくなるほど転送時間の差異は大きくなる。

本節で作成したデータ通信方式を利用することで、同一の計算機を利用する場合のシミュレーション時の計算時間の削減を可能とした。データの転送が同一計算機上で行われる場合は共有メモリを使用し、転送がネットワークを介して行われる場合は既存の CORBA 通信を行うデータポートを実装した。故に作成したデータポートクラスを使用することで、いくつかのモデルを結合させるような単純なモデルシミュレーションをしようとする場合において、シミュレーション時間の短縮が見込まれる。そして結合させるモデルが増加(モデルが大規模化)し、シミュレーションの計算時間が増加した場合は、複数の計算機と既存の数進奉仕を利用することで、計算負荷の分散を行わせることができる。しかし計算機への負荷分散は現在手動で行わなければならない、自動負荷分散の実装が今後の課題となる。

4.4. コンポーネント化システム

OpenRTM-aist 上での一般的なモデル構築手順について説明する。Windows 上で OpenCV の関数をコンポーネント化する場合、OpenRTM-aist では一般的に以下の手順で

コンポーネントを構築する。

- (1) RTCBuilder を使用したテンプレート作成
 - (1-1) プロジェクトを新規作成
 - (1-2) 基本設定(コンポーネント名などを指定)
 - (1-3) 状態遷移設定(使用する状態遷移)
 - (1-4) ポート設定(ポート名・データ型・変数名などを指定)
 - (1-5) パラメータ設定(パラメータ名・データ型・変数名などを指定)
 - (1-6) 使用言語設定(C++, Python, Java など)
 - (1-7) コードを生成
- (2) CMake(GUI)でのコンパイル
 - (2-1) CMakeLists.txt の編集(必要ライブラリのパスを追加)
 - (2-2) コンパイルするプロジェクトを指定
 - (2-3) コンパイラを指定
 - (2-4) コンパイルを実行
- (3) Visual Studio 上でのプログラムの記述
 - (3-1) コンパイルしたプロジェクトを開く
 - (3-2) ヘッダファイルの編集(必要ライブラリのインクルード, 変数宣言など)

(3-3) ソースファイルの編集(処理内容の記述)

(3-4) コンポーネントのビルド

これら手順を踏むことで、コンポーネントを一つ作成することができる。少なくとも3種類のソフトウェアを使用する必要があり、非常に手間であることがわかる。再利用したい既存プログラムがごく少数であるならば問題はない。しかし実際には実装対象である数理モデルや既存ライブラリは多数存在している。例えば、疎粒度モデルの構築で広く使用されている OpenCV では入力画像データであるものに限っても、39 個以上の関数やアルゴリズムが存在する(OpenCV 2.4.5)。今後さらに数理モデルが提案されることや、バージョンアップによる OpenCV 関数の増加を考慮する必要がある。以上の理由から、手作業でのコンポーネント構築では要件 III-I 及び IV-III を満たさないことがわかる。

そこで本研究では、既存の数理モデルやライブラリをコンポーネント化するためのソフトウェア OpenCV-RTC を開発し、HI-brain の公式サイトで公開した。OpenCV-RTC を使用することで OpenCV 関数のコンポーネントを容易に利用でき、使用者ごとに使用したいモデルやライブラリの関数を追加することができる。OpenCV 関数をコンポーネント化するために、OpenRTM-aist のコンポーネントと OpenCV 関数には共通構造があることに着目した(Fig. 19)。さらに、これらの共通構造を実体化するプログラミング規約が OpenRTM-aist には存在しているため[11]、OpenCV 関数とコンポーネントは共通構造

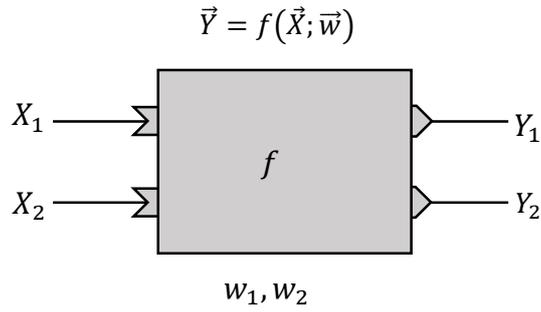


Fig. 19: 画像処理関数や視覚モデル, RT コンポーネントの共通構造. 入力データは $\vec{X} = (X_1, X_2)$, 出力データは \vec{Y} , パラメータは \vec{w} で表す. 関数 f は $(\vec{X}; \vec{w}) \mapsto \vec{Y}$ の写像を意味する.

を介した相互変換を可能とする.

OpenCV の各画像処理関数は以下の数式で表すことができる.

$$\begin{aligned}
 \vec{Y} &= f(\vec{X}; \vec{W}) \\
 \vec{X} &= (X_1, X_2, \dots, X_l) \\
 \vec{Y} &= (Y_1, Y_2, \dots, Y_m) \\
 \vec{W} &= (w_1, w_2, \dots, w_n)
 \end{aligned}
 \tag{9}$$

関数 f は入力ベクトル \vec{X} 及びパラメータベクトル \vec{W} を引数とし, \vec{Y} を出力する. それぞれのベクトルの長さは関数ごとに異なるが, 画像処理の関数はこの形式と一致する. 一方

Table 3: OpenRTM-aist と画像処理関数の対応関係. OpenRTM-aist のコンポーネントに関数を記述する場合の, 関数の記述個所と引数の指定方法を表す.

OpenRTM-aist の記述規約	関数及び関数の引数
onExecute()	f
bindParameter()	\vec{W}
inPort()	\vec{X}
outPort()	\vec{Y}

OpenRTM-aist では、OpenRTM-aist の関数である「`addInPort()`・`onExecute()`・`bindParameter()`・`addOutPort()`」によって格納された変数の役割が指定される[11]. `addInPort()`は入力ポートと呼ばれる、コンポーネントが情報を受信するためのインターフェースを追加する関数である. 同様に `addOutPort()`は出力ポートを追加する関数である. `bindParameter()`は指定された変数がパラメータであることを指示する関数である. 遷移する状態の一つである `onExecute()`内に関数を記述することで、周期ごとにその関数が実行される. よって画像処理関数の構造と、OpenRTM-aist の関数是一对一対応していることがわかる(Table 3). OpenCV-RTC は、この対応関係をマッピングするソフトウェアである.

OpenCV-RTC は「定義ファイル」を基にコンポーネントを生成する. この定義ファイルは関数及びその引数とコンポーネント間の対応関係を記述したものである. 例えば、式(9)の関数 f が、入力データ $\vec{X} = (X_1, X_2)$ 、出力データ $\vec{Y} = (y_1, y_2)$ 、パラメータ値 $\vec{W} = (w_1, w_2)$ を持つ場合を考える. このときの変数型はそれぞれ `typeX`, `typeY`, `int` とする. この関数の C++言語での関数宣言は以下のコードで与えられる.

1	<code>void f(</code>	
2	<code>const typeX&</code>	<code>X1,</code>
3	<code>const typeX&</code>	<code>X2,</code>
4	<code>int</code>	<code>w1,</code>
5	<code>int</code>	<code>w2,</code>

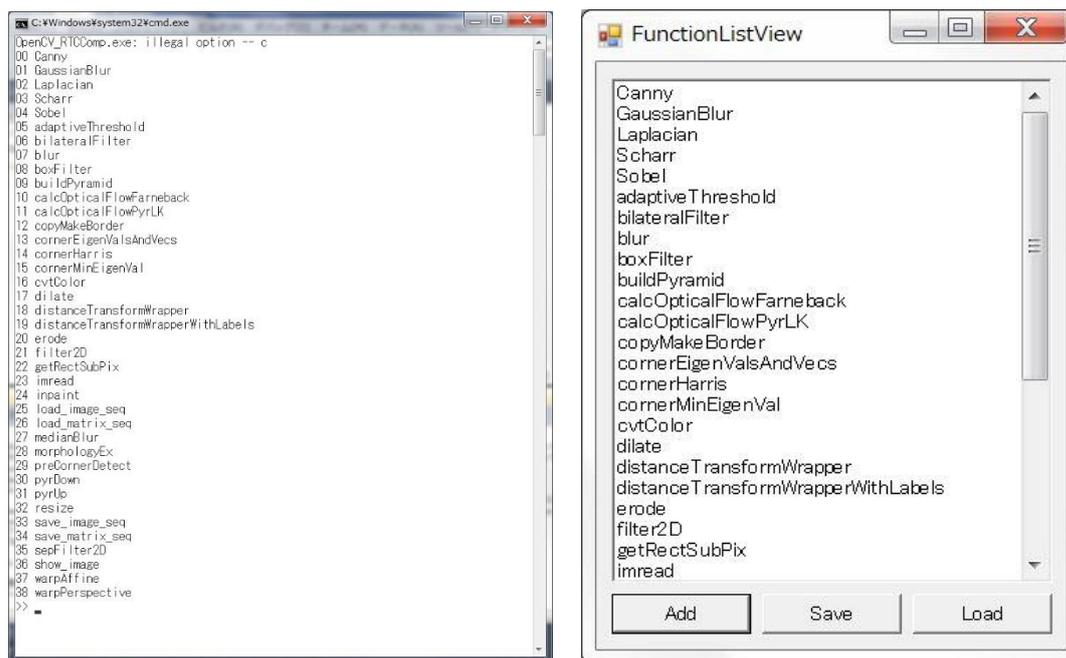
```
6 typeY&          Y1,  
7 typeY&          Y2  
8 );
```

この関数をコンポーネント化するための定義ファイルは次のとおりである。

```
1 void@unused f(  
2 typeX          X1@inport,  
3 typeX          X2@inport,  
4 int            w1@param,  
5 int            w2@param,  
6 typeY &        Y1@outport,  
7 typeY &        Y2@outport  
8 );
```

C++の関数宣言と類似した形式であるが、定義ファイルの場合は変数名の後に@を挟んで役割を記述する必要がある。例えば、X1は@inportを追記することで入力データとして使用することを意味する。同様に@outportは出力データ、@paramはパラメータを意味する。関数*f*がOpenCV関数である場合、typeXやtypeYはcv::MatなどのOpenCVのデータフォーマットを記述する。

OpenCV-RTCにはCUIとGUIの2種類の起動用バッチファイルが用意されており、これを起動することで定義ファイルに記述した関数のリストが表示される(Fig. 20)。そこから使用したい関数を選択することで、その関数のコンポーネントが作成される。定義ファイル内には複数の関数を記述することができるため、OpenCV-RTCから多種多様



(A) CUI で起動した場合

(B) GUI で起動した場合

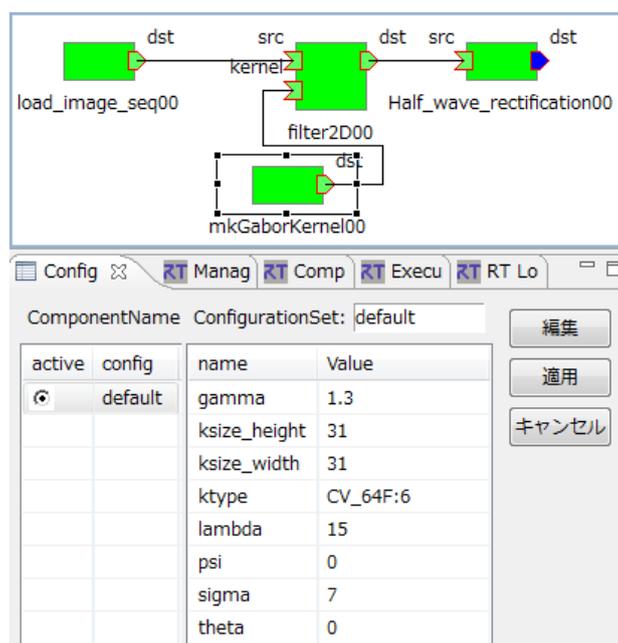
Fig. 20: OpenCV-RTC で表示される関数リスト

な関数を使用することができる。また OpenCV-RTC は標準で OpenCV の画像処理関数を用意しているため、OpenCV-RTC を実行するだけで既存の関数を容易に利用することができる(Table 4)。このことから要件 III-I を満たしていると考えられる。

OpenCV-RTC を使用した疎粒度モデルの構築例として、V1「単純型」細胞のモデルを OpenRTM-aist 上で構築した(Fig. 21)。脳の V1 野と呼ばれる脳領域には、物体の方位に対して選択的に反応する細胞が存在する。これを画像処理的の関数として記述する場合、エッジ抽出とその結果の非線形変換によってモデル化することができる。

このモデルを OpenCV で実装する場合、以下の 3 種類の関数を用いる。

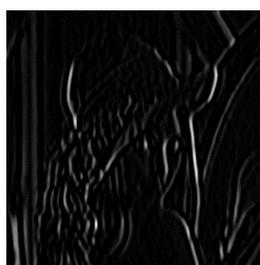
(1) フィルタ演算を行う関数



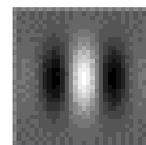
(A) V1 単純型細胞のモデルの結合図



(B) 入力画像



(C) 出力画像



(D) V1 受容野

Fig. 21: (A) OpenRTM-aist 上で構築した V1 単純型細胞のモデル. 下部は mkGaborKernel のパラメータを表示している. σ や θ のパラメータを変化させることで様々な V1 の受容野を生成する. (D) 垂直なエッジに対して反応する V1 の受容野. θ を 0 に設定することで同一のカーネルが生成できる.

(2) フィルタの定義

(3) フィルタリング結果の非線形変換

これら 3 種類の計算を行う OpenCV 関数を OpenCV-RTC で実装, そのシミュレーシ

ョンを行った。フィルタリング処理を行うには、OpenCV 関数の一つである `cv::filter2D()` を使用することで実行できる。以下のコードは C++ で実装された `cv::filter2D()` の関数宣言である。

```
1 void filter2D(  
2  const cv::Mat&      src,  
3  cv::Mat&           dst,  
4  int                 ddepth,  
5  Const cv::Mat&     kernel,  
6  cv::Point           anchor = cv::Point(-1,-1),  
7  double              delta = 0.0,  
8  int                 borderType = BORDER_DEFAULT  
9  );
```

第 1 引数の `src` は入力画像、第 4 引数の `kernel` はフィルタ関数を指定するものであり、これを入力ポートに割り付けることで、様々なフィルタでの計算が可能である。第 2 引数 `dst` は出力画像であり、この関数の出力が格納される。その他の引数のフィルタリングに関するパラメータである。この関数を OpenCV-RTC で実行するために、以下の以下の通りに定義ファイルを記述する必要がある。

```
1 void@unused filter2D(  
2  cv::Mat          src@inport,  
3  cv::Mat&        dst@outport,  
4  int              ddepth@param=-1,
```

```

5 cv::Mat&          kernel@inport,
6 cv::Point        anchor@param=cv::Point(-1,-1),
7 double           delta@param=0.0,
8 int              borderType@list/borderType=BORDER_DEFAULT
9 );

```

第 1 引数である `src` と、第 4 引数である `kernel` はコンポーネントの外部からデータを与えるために、`@inport` 属性を付与する。これによって、コンポーネントの入力ポートとして割り付けられる(Fig. 21-A 参照)。第 2 引数の `dst` はこの関数の出力が格納されるため、出力ポートとしたい。そのため `@outpot` 属性を付与した。`@param` または `@list` 属性が付いた引数はパラメータである。パラメータに既定値を与えたい場合は第 6 引数の `delta@param=0.0` のようにして与えることができる。

`cv::filter2D()` ではフィルタの内容を外部から与える必要がある。V1 単純型細胞の場合、フィルタ関数として Gabor 関数を使用することが一般的である。フィルタの定義では、OpenCV 関数である `cv::getGaborKernel()` を用いた独自関数 `mkGaborKernel()` 関数を作成し、OpenRTM-aist のコンポーネントに変換した；

```

1 void mkGaborKernel(cv::Mat& dst, cv::Size ksize, double sigma,
   double theta, double lambda, double gamma, double psi, int ktype)
2 {
3   dst = cv::getGaborKernel(ksize, sigma, theta, lambda, gamma, psi,
   ktype);

```

```
4 };
```

フィルタリング後に非線形変換を行った結果が、V1 単純型細胞のモデルの出力となる。半波整流関数はその代表的な関数であり、これは OpenCV の `cv::max()`関数を使用した `Half_wave_rectification()`関数を作成し、OpenCV-RTC を使用してコンポーネント化した。

```
1 void Half_wave_rectification(cv::Mat& src, double threshold,
2 cv::Mat& dst)
3 {
4   cv::max(src-threshold,0,dst);
5 };
```

このように個々の OpenCV 関数を直接又は使用した関数を作成しコンポーネント化を行うことで、これらをモデルの構成要素としてモデルを構築し、シミュレーションすることができる。その他、OpenRTM-aist のコンポーネントとして実行可能な OpenCV 関数の一覧を Table 4 に示す。

要件 III-I を満たすための追加機能として、OpenCV-RTC を開発した。OpenCV-RTC は OpenCV 関数と同様に、既存のモデルや関数を本基盤に対応したコンポーネントに容易に変換することができる。これによって今後開発されるであろう新規モデルや新規自作関数への対応も可能である。この作成手順は一般的な OpenRTM-aist のコンポーネント作成手順と比べ容易にコンポーネントを作成することができるため、要件 IV-III も同様

に満たしていることがわかる。

Table 4: OpenCV-RTC が提供する関数一覧。

関数名	視覚モデルとの対応など
bilateralFilter	画像フィルタの一種. エッジを保存しつつノイズを除去する. V1 野における情報処理との関連性が強い.
blur	ぼかし関数.
boxFilter	もっとも単純な平滑化フィルタカーネル. 細胞間結合に使用できる.
buildPyramid	多解像度ピラミッドを構成する関数. 視覚系が多解像度処理を行っているため, モデル化に有用な関数.
copyMakeBorder	与えられた画像を上下左右にコピーする関数. 視覚モデルでは畳み込みが多用されるのだが, 本関数を利用すると畳み込みの境界処理が容易になる.
dilate	モルフォロジー変換の一つ. 画像膨張を行う関数 (単純な縮小ではない). 視覚系との対応は今のところ不明であるが, 文字認識時など高次視覚情報処理のモデル化に有用である可能性がある.
erode	モルフォロジー変換の一つ. 画像縮小を行う関数.
filter2D	一般的な 2 次元フィルタを構成する関数. この関数を通じて適切な値を設定することで, 様々な四角細胞モデルを作成することができる.
GaussianBlur	ガウスぼかしを行うフィルタ. 細胞間結合強度を与える関数としても有用.
medianBlur	中央値を返す関数. 視覚系との対応は現在のところ見出されていないが, 将来的に脳計算論との対応が見出されるかもしれない.
morphologyEx	上記 dilate と erode 以外のモルフォロジー変換を行う関数.
Laplacian	ラプラシアン関数. 微分フィルタの一種であり, 網膜神経節細胞の受容野関数の一つとして用いられることが多い.
pyrDown	画像の平滑化とダウンサンプリングを行う関数. 多解像度ピラミッドを任意の階層で構成する再にも有用.
pyrUp	画像のアップサンプリングと平滑化. 上記と逆.
sepFilter2D	分離可能フィルタを構成する関数. V1 以前の数理モデルを構築する再にも有用.
Sobel	微分フィルタの一種. 画像微分を用いた数理モデルは多数提案されている. MT/MST モデルは本関数を何度も使用する.
Scharr	微分フィルタの一種. Sobel と同様のフィルタであるが, 特性が若干異なる. 視覚モデルに使われた例は筆者の知る限りない.
getRectSubPix	サブピクセル値を得る関数. 視覚系は, 網膜光受容細胞の密度から予測されるよりも高い解像度で, 「ずれ」を見出すことが知られている. これを説明する最も単純な関数.
resize	画像サイズの変更.
warpAffine	画像のアフィン変換 (平行移動, 拡大縮小, 回転) を行う関数. 入力画像生成や前処理に用いられる.
WarpPerspective	射影変換. 入力画像生成時に用いられる. 外界像から網膜像を得るための基本的な関数.

関数名	視覚モデルとの対応など
adaptiveThreshold	適応的閾値処理. 視覚系でも画像の輝度などに応じて閾値を変化させた画像処理を行っていることが知られている.
cvtColor	色空間の変換. RGB から Luv 色空間への変換などをおこなう. そのほか適切なパラメータを設定することで, 視覚系固有の色空間を得ることも原理的には可能.
distanceTransformWrapper	OpenCV 2.4.5 関数の distanceTransform (const Mat&, Mat&, int, int) を Wrap した OpenCV-RTC 独自関数
distanceTransformWrapperWithLabels	OpenCV 2.4.5 関数の distanceTransform (const Mat&, Mat&, Mat&, int, int, int) を Wrap した OpenCV-RTC 独自関数
Canny	微分フィルタの一種.
cornerEigenValsAndVecs	画像中のコーナーを検出するために利用する関数. コーナー以外の画像は, いわゆる窓問題が発生することを意味するため, 運動視 MT/MST モデル構築時の基礎的な関数として利用できる.
cornerHarris	コーナー検出関数の一種.
cornerMinEigenVal	コーナー検出の際に導出される 2 つの固有値のうち小さい固有値を求める関数. この固有値の大小で, 窓問題が発生しているかどうかを判定することができる.
preCornerDetect	コーナー検出器のための特徴マップを得る関数. 視覚系との対応は調査中.
show_image	OpenCV 関数の imshow (const string&, const Mat&) を Wrap した OpenCV-RTC 独自関数
imread	画像読み込み関数
inpaint	画像充填関数. 盲点情報処理の計算原理として利用される. 神経細胞との対応もよい.
calcOpticalFlowPyrLK	Lucas-Kanade 法に基づくオプティカルフロー計算関数. MT 細胞の計算原理として利用される.
calcOpticalFlowFarneback	Farneback 法に基づくオプティカルフロー計算関数.
load_image_seq	画像イメージファイルを連続して読み込むことができる OpenCV-RTC 独自関数
save_image_seq	画像イメージファイルを連続して書き出すことができる OpenCV-RTC 独自関数
load_matrix_seq	画像マトリクスファイルを連続して読み込むことができる OpenCV-RTC 独自関数)
save_matrix_seq	画像マトリクスファイルを連続して書き出すことができる OpenCV-RTC 独自関数

4.5. モデル用データベース

これまでの機能拡張によって, 本基盤で作成したモデルは互換性を持たせることがで

きたが、それだけでは協調的モデル研究を行うことは難しい。モデルの大規模化には、モデルが再現する脳領域や脳機能に関する知見や作成に費やす労力を必要とする。そのため個人での大規模モデルの実装は難しく、多くのモデル研究者は小規模なモデル研究に終始してきた。作成したモデルを研究者間で共有することができれば、新規モデルや大規模視覚数理モデルをゼロから作成する必要はない。既存モデルをベースに新規モデルの開発を行なうことは、既存モデルと比較してどのようにそのモデルが優れているかがわかりやすくなるというメリットがある。本節では本基盤で作成したモデルを共有するためのウェブサイトとデータベースを作成した。

数理モデルに関するデータベースはいくつか存在している^{12,13}。しかし既存のデータベースでは、公開されているモデルは開発環境(言語やライブラリ、フォーマットなど)が異なっているため、モデルの再利用や改良が難しい。また本基盤で作成したモデルコンポーネントをロボット研究者が利用することも期待しているため、数理モデル用データベースでは難しいと判断した。

作成するデータベースは本基盤で作成した数理モデルの再利用だけでなく、既存モデルの改良も目的とする。そのため、データベース上のモデルが何のモデルを基に、どのような改良を行ったかなどの情報が必要となる。これを行うには一般的なデータベース

¹² <https://visiome.neuroinf.jp/>

¹³ <https://senselab.med.yale.edu/modeldb/>

のようなダウンロードとアップロード機能だけでは難しい。そこでバージョン管理システムである Git と共有ウェブサービスである Github を利用した。

Github の機能を利用することで、上記問題の解決を図ることができる。Github はバージョン管理システムである Git を使用しているため、モデル自体をアップロードするだけでなく、変更履歴も確認することができる。そのため、視覚モデルの管理だけでなくモデル開発の管理を行うことができる。また、モデルのアップロード権限を他者に与えることができ、他者と一つのモデルを開発することができる。また Github は視覚研究者だけではなく広く利用されているため、モデルの他分野への応用が広く期待できる。

Github API を利用して、HI-brain のウェブサイトを構築した¹⁴。ウェブサイトの機能の一つが Github API を利用した HI-brain 形式のモデル一覧である。研究者が Github アカウントで HI-brain サイトにログインすることで、研究者が Github に挙げているモデルが、ウェブサイトのモデル一覧に追加される。その一覧から必要となるモデルを探し、ダウンロードすることで互換性を持ったモデルを使用することができる。

4.6. データフォーマットの相互変換ライブラリ

要件 IV-I を満たすために新たに開発したシステムについて説明する。4.1 節で作成した疎粒度モデル用データ型である `Timed_cvMat` 型を使用することでモデル間の結合を可能とする。しかしコンポーネントの内部処理で使用されている関数には、`Timed_cvMat`

¹⁴ <http://hi-brain.org/>

型を使用することはできない。使用されている関数が OpenCV である場合は `cv::Mat` へ、MATLAB 関数である場合は `MATLAB::mat` へ変換する必要がある。そのため、`cv::Mat` を中心としたデータフォーマット・データ型の形式変換ライブラリを作成した。

現在このライブラリでは以下の変換が可能である。

- `cv::Mat` ⇔ `Timed_cvMat` 型
- `cv::Mat` ⇔ `MATLAB::mat`
- `cv::Mat` ⇔ `CameraImage` 型

`cv::Mat` は OpenCV の画像処理関数で利用される配列フォーマットであり、`Timed_cvMat` は疎粒度モデルのコンポーネントの入出力データを格納するデータ型である。そのため疎粒度モデルをコンポーネントとして記述する際には `cv::Mat` ⇔ `Timed_cvMat` 型の変換が必要となる。また MATLAB は疎粒度モデル開発の主要言語であるため、MATLAB で記述された関数を OpenRTM-aist のコンポーネントとして使用するには MATLAB Compiler を使用した C++ライブラリへの変換が必要となる。その際に使用するフォーマットが `MATLAB::mat` であり、これと `cv::Mat` への変換を容易とすることで、MATLAB で記述された既存モデルを容易にコンポーネント化することができる。`CameraImage` 型は OpenRTM-aist で用意されている画像用データ型である。OpenRTM-aist でロボット開発の際に広く利用されているため、`cv::Mat` と `CameraImage` 型の相互変換は既存コンポーネントを使用する際に必要となるだろう。

疎粒度モデルの開発で使用されるデータフォーマット・データ型への相互変換ライブラリを作成した。ライブラリを使用することにより、OpenCV 関数をコンポーネントした際のデータの形式変換の手間を減らすことができる。現状では、`cv::Mat`, `Timed_cvMat` 型, `MATLAB::mat`, `CameraImage` 型を対象としているが、`cv::Mat` を中心とすることで、どのようなデータフォーマット・データ型であっても、`cv::Mat` に変換することができる。例えば、`Timed_cvMat` 型, `MATLAB::mat`, `CameraImage` 型へ容易に変換することができる。このライブラリは 4.5 節で作成した HI-brain の公式サイトおよび Github¹⁵で公開している。

4.7. 容易な環境実装方法

HI-brain は OpenRTM-aist や OpenCV などの既存のソフトウェアを基にモデル研究用の機能を追加開発したものである。これら環境を手作業で構築するのは手間と時間がかかる。Windows(64bit)に構築する場合は、Java JRE, python, python(64bit), PyYAML, OpenRTM-aist, VisualStudio, cmake, doxygen, OpenRTP, OpenCV, Matlab Compiler Runtime などのソフトウェアの適切な事前設定とインストールを必要とする。この問題点を解決しなければ、HI-brain をモデル研究者に対して普及させることは困難であると考えられる。そこで要件 IV-II を満たすための追加開発として、HI-brain の環境を実装した Linux OS を作成し、HI-OS と名付けた。

¹⁵ <https://github.com/hi-brain/>

HI-OS は Linux 環境の一般的なインストール手順に従うことで導入される。具体的には、(1) HI-OS を HI-brain のウェブサイトからダウンロード、(2) 物理マシン又は仮想マシン (VMware Player¹⁶, Oracle VM VirtualBox¹⁷ など) 上にインストールすることで使用することができる。インストール方法の詳細は HI-brain のウェブサイトに記載している。このように手作業で構築する場合と比較して、より簡易で確実に環境を利用することができる。作成した OS は、メジャーな Linux である Ubuntu¹⁸ を基にしているため、Linux が未経験者であっても比較的問題を解決しやすい。またすでに Linux 環境を持つ研究者向けに HI-brain 環境をインストールするシェルスクリプトを作成し、HI-OS 同様公開している。

OS から環境を用意する手法は、これまでも多く存在している。例えば、高エネルギー物理学などを主な目的とした Scientific Linux¹⁹ がある。またオープンソースの数学ソフトウェアを実装させた数学者向け OS、MathLibre²⁰ がある。このように用途、対象を限定したディストリビューションをあらかじめ作成することで容易にソフトウェア基盤を使用することができる。HI-OS もまた視覚モデル研究者向け OS として公開し、視覚モデル研究者共通の環境として使用されることを期待する。

¹⁶ <http://www.vmware.com/>

¹⁷ <http://www.oracle.com/jp/index.html>

¹⁸ <https://www.ubuntu.com/>

¹⁹ <https://www.scientificlinux.org/>

²⁰ <http://www.mathlibre.org/index.html>

4.8. まとめ

4章では, OpenRTM-aist で疎粒度モデルを開発する際に必要となる機能を追加開発した. 開発した機能は 3 章で述べた必要要件を満たし, OpenRTM-aist を疎粒度モデル開発用基盤として拡張した.

必要要件 I-II 及び III-I を満たすため, 4.1 節では疎粒度モデルで広く使用されている画像処理ライブラリである OpenCV のデータフォーマットを基に疎粒度モデル用データ型を作成した. 4.2 節では結合させたモデルの同期実行機能を構築し, 正確なモデルシミュレーションのためにモデルの実行周期を管理する. これは必要要件 II-I を満たした. 4.3 節では共有メモリを使用使用したデータ通信方式を作成, 必要要件 II-III である高速データ通信を可能とした. また 4.4 節では OpenCV を含むライブラリの各種計算アルゴリズム及び既存モデルを, OpenRTM-aist の規約に従って呼び出すソフトウェアを作成し, 必要要件 III-I 及び IV-III を満たした. これらを使用することで, 既存モデルを疎粒度モデルのデータ型を入出力するコンポーネントに変換し, モデル間の結合・置換を容易にする, 4.5 節以降は主にソフトウェア基盤の利便性向上のための機能開発であり, モデル開発の手間を省く. 4.5 節は他研究者とのモデル共有のためのモデルデータベースであり, 本基盤で構築されたモデルを共有する. 4.6 節はデータフォーマット・データ型間の形式変換を自動化するためのライブラリであり, モデルコンポーネントを記述する際に必要となる「コンポーネントの入力データ型から入力データフォーマット

への変換」及び「関数の出力データフォーマットからコンポーネントの出力データ型への変換」を省略することができる。

これら追加機能を使用することで OpenRTM-aist で疎粒度モデル開発が可能となる。

しかし OpenRTM-aist にはモデル開発を行なった実績がないため、可能であることを証明する必要がある。次章では実際に疎粒度モデル開発を行なうことで、本基盤で疎粒度モデルのコンポーネント指向開発が可能であることを証明する。また本基盤の利点には、様々な環境に対応していることや、並列分散処理が可能であることも挙げられる。これら利点を用いたモデル開発例もいくつか紹介する。

5. モデル構築の具体例と結果

前章にて視覚数理モデル開発用基盤としての必要要件を満たすための追加機能を構築した。疎粒度の視覚数理モデルは、脳領野または脳機能をモデル化の単位としている。そのため全脳領野とその機能を網羅するように、個々のモデルを直列・並列に結合させることで視覚全体のモデル化とシミュレーションができるのではないかと考えられる。しかし現実性がないため、それを可能にするには HI-brain のようなモデルの結合や、モデルの一部改良・一部修正を容易にするソフトウェア基盤が必要であった。本章では実際のモデル開発を例に、共通ソフトウェア基盤の重要性を示す。また学術的に新規性の高い知見をもたらす数理モデルが本基盤上で構築可能であることを示す。5.1 節では器官モデルである網膜のモデルと高次脳機能モデルとの結合を行った。多くのモデルで考慮されていない網膜モデルが高次機能へどのような影響を与えるかを示し、モデル間結合の重要性を示す。5.2 節では提案されている同種視覚モデルを結合させた、新規モデル開発の手法を示す。既存モデルを複数組み合わせることで、新たなモデルを開発することができることを証明する。3 節は速度選択性モデルを例に既存モデルを再利用し、新たな知見に基づいた改良を行った。新たな知見が現れるたびに新たなモデルを開発するのではなく、既存モデルへの改良を施すことで、既存モデルとの違いが明確となる。5.4 節ではモデルの実装例として、小型計算機である RaspberryPi を複数用意し、それぞれ異なるモデルを実装し結合させた。5.5 節では HI-brain の利点の一つである、異なる

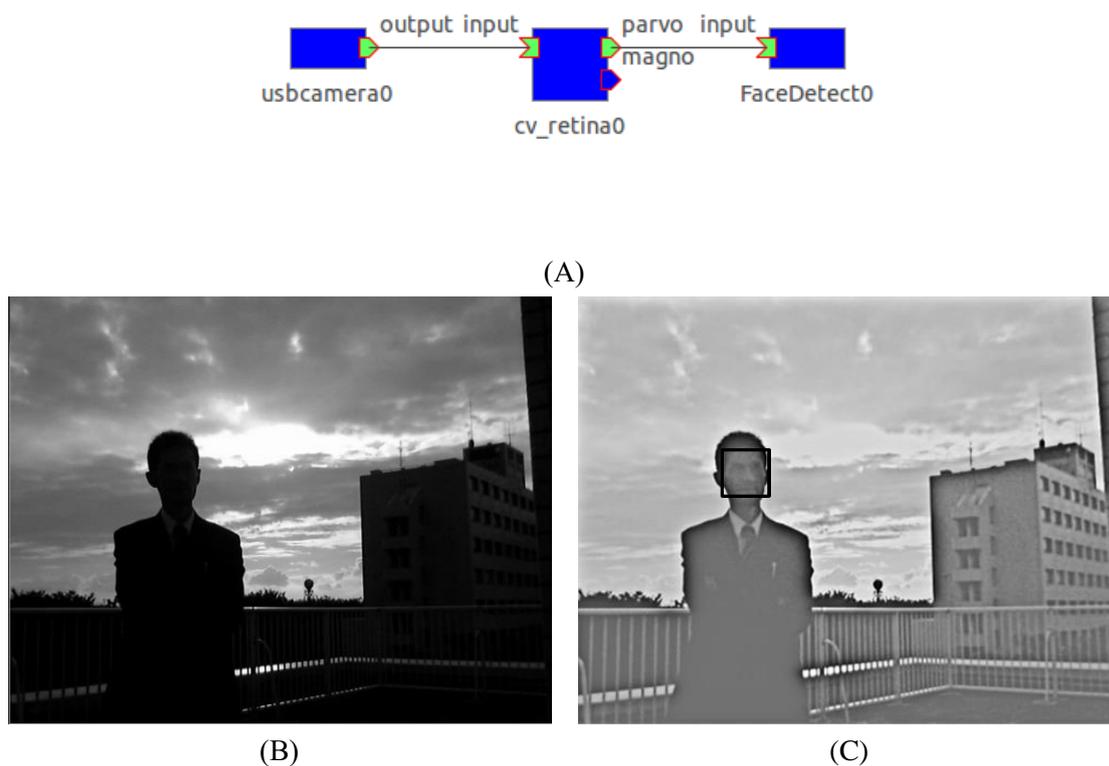


Fig. 22: 低次の器官モデルと高次視覚機能モデルとの結合例. (A) OpenRTM-aist 上で網膜モデルと顔検出モデルを結合させた様子. (B) USB カメラの画像と顔検出の結果. (C) 網膜モデルの Parvo 系の出力画像と顔検出の結果.

計算機・OS 環境上で実装したモデル間の結合が可能であることを示す.

5.1. 器官モデルと脳機能モデル間の結合

網膜モデルと高次脳機能の一つである顔検出のモデルと組み合わせることで、工学的に有用な結果が得られることを示す. (Fig. 22). 「網膜は脳の出先器官である[46]」と言われるほどに複雑な細胞網によって形成されており、適応的局所ゲインコントロールや色空間変換などの画像処理を実行している. そのため、網膜上で行われた画像処理結果が脳で構築されるすべての視覚機能に影響を与える. 視覚システムを正しく理解するに

はこれを考慮しなければならない。そのため網膜モデルを容易に実行できる環境があれば多くの研究開発者に利点をもたらすと考えられる。そこで4.2節で説明した OpenCV-RTC を利用し、OpenCV ライブラリのクラスの一つである `cv::Retina` を OpenRTM-aist 上で実行できるコンポーネントとして実装した。

ヒトの網膜は視細胞、水平細胞、双極細胞、アマクリン細胞、神経節細胞で構成されており、神経節細胞からの出力が網膜の最終的な出力となる。これら様々な細胞によって構成される網膜情報処理によって、網膜上では適応的なコントラストゲインコントロール（輝度の低い領域を明るく、かつ細部をはっきりさせる）や、時空間ノイズの除去、輪郭強調が行われていると考えられている。`cv::Retina` は網膜の最終的な出力である 2 種類の神経節細胞を出力する。神経節細胞は大細胞系（*magno*）と小細胞系（*parvo*）に大別され、*magno* 系細胞は画像の時間的変化を符号化し、*parvo* 系では適応的なコントラストゲインコントロールを実行している[47]。

Fig. 22-B,C に USB カメラで撮影された像と、網膜モデルの 2 種類の出力の一つである *parvo* 系の出力結果を示す。左の RTC は USB カメラからの画像の取得を行う。中央の RTC は網膜の *Magno* 系と *Parvo* 系細胞のシミュレーションを行う。Fig. 22-B は逆光下で写真撮影をした結果、人物の顔や建造物、鉄柵の細かい模様が黒つぶれしていることがわかる。一方、網膜出力細胞の 1 種である *parvo* 系細胞の出力を見ると、カメラ画像と比較すると黒つぶれした位置の 패턴の視認性が向上していることがわかる。実

際、筆者の主観的な評価では網膜モデルの出力の方が実際の見えと類似していた。この機能によって人は、日常環境下の暗所にも明所にも柔軟に適応する画像処理を行っている。一般的に構築困難な視覚モデルを、脳科学研究者でなくとも利用できる環境の構築は多くの研究にとって重要であると筆者らは考えている。magno 系細胞の有効な工学的利用方法については今後の課題とする。

高次視覚機能の例として、高次視覚機能である顔検出のモデルを作成し網膜モデルとの結合を行った(Fig. 22-A)。顔検出モデルは `cv::CascadeClassifier` クラスを使用して作成した。カメラで撮影した元画像では顔を検出することは出来ず、網膜 parvo 系の出力では顔を検出することが出来た (Fig. 22-B,C)。実際の視覚と同様に脳の処理を行う前に網膜の処理を行うことでより正確な結果が得られる例を示し、モデル結合とそれを可能にするソフトウェア基盤の有用性を示した。また、このように数理モデルの挿入や削除を容易に行えるソフトウェア基盤は、脳神経系の研究のみならず、工学的な応用としても有効であると考えられる。

5.2. 新規性の高い数理モデルの構築

5.2.1. 注視モデルの概要

3.2 節で説明した注視モデルは、これまでさまざまな計算理論や実験結果に基づいて提案されてきた。このモデルはある画像を入力として与えると、Fig. 23 のような顕著度マップ(Saliency map)と呼ばれる画像または 2 次元配列を出力する。Fig. 23 は赤い箇所

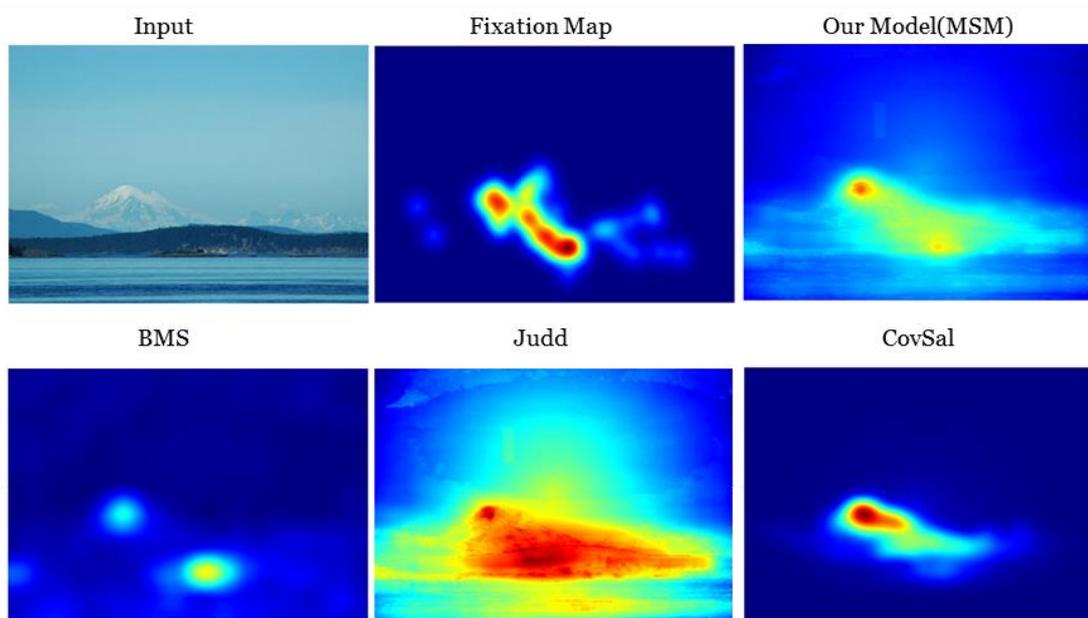


Fig. 23: 入力画像とヒトの注視マップに対する既存モデルや新規モデルの顕著度マップ.

ほど顕著性が高い、つまりヒトがそこを注視しやすいと予測した箇所であり、逆に青い箇所ほど顕著性が低く、ヒトがそこに注意を向けにくい箇所であることを表している。

ヒトが実際にどの箇所を注視するかを調べるために、眼球運動測定装置によって測定さ

れた注視マップ(fixation map)と呼ばれる2次元配列を作成する。最も精度の高いモデルは、これに近い顕著度マップを出力する。注視マップは注視された回数が多いまたは長く注視された箇所を赤で、逆に注視されなかった箇所を青で可視化している(Fig. 23)。

Saliency map モデルは実際にヒトが注視した結果である注視マップと同じ顕著度マップをモデルの出力として得ることである。モデルの優劣は MIT Saliency benchmark におい

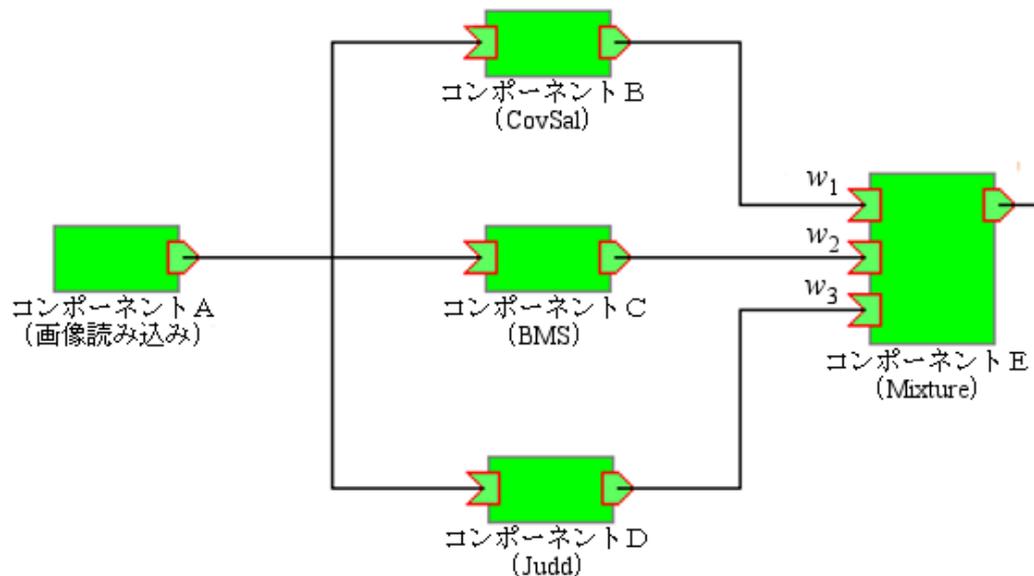


Fig. 24: 作成した新規注視モデルの構造.

て定量的に評価されており、モデル数は30を超えている²¹。

本節では、視覚的注視位置を予測する新規モデルが、既存注視モデルの結合によって構築でき(Fig. 24)、その定量評価値も優れていることを示す(Fig. 25)。

5.2.2. 既存モデルの再利用によるモデル構築

既存モデルの性能を超えるモデルを構築する場合、心理物理実験の結果をもとに新たな理論を構築することが一般的である。しかし、既存の注視モデルを結合、出力を合成させるだけで、より良いモデルを構築できないだろうか？そこでHI-brainの機能を用い

²¹ <http://saliency.mit.edu/>

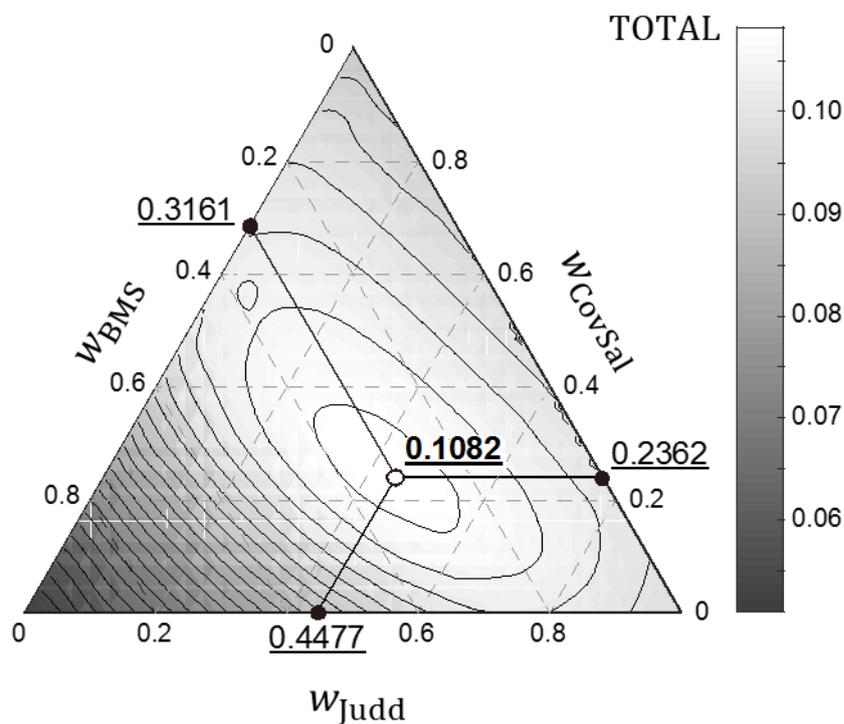


Fig. 25: それぞれの重みに対する新規モデルの評価値のプロット図. 3 角形の各端の色はモデル単体での結果を示し, 3 角形の中心だと重みが等しい場合での結果を表している. 重みが $\vec{w} = \{0.2362, 0.3161, 0.4477\}$ の場合, 最大の評価値である 0.1082 を得る.

ることで, 複数の注視モデルに重みを加え, 足し合わせることで最も性能の良いモデルが作れるのではないかと考えた.

上記のモデル構築の有用性を確認するために, 2014 年 4 月の段階で MIT benchmark において高い評価値を得ていた 3 種の既存数理モデル: CovSal[48], BMS[49], Judd[50] をコンポーネント化し, OpenRTM-aist 上で実行できるようにした. 各モデルの入出力データ型には 4.1 節の Timed_cvMat を使用し, モデル結合や出力の合成を可能とした. ま

たこれら 3 種類のモデルは MATLAB 言語で記述されていた。そのため MATLAB 言語で用いられている行列形式 MATLAB::mat を Timed_cvMat 型に変換するライブラリを使用した

Fig. 24 に、3 つのモデルの線型和で表現される新規モデルを OpenRTM-aist 上で実行している様子を示す。

5.2.3. 新規注視モデルと定量評価

3 種類の注視モデルを結合させた新規モデルの定量評価を行った。Fig. 25 に新規モデルの定量評価値を示す。Fig. 25 は各モデルに対する重みとその評価値を色の濃淡で表したものである。色が白に近いほど評価値が高く、逆に黒に近い色ほど評価値が低いことを意味する。このことから Fig. 25 は上に凸の単峰性面であることがわかった。また適切な線形重みを設定すると既存モデルの性能を超えることがわかった。具体的には 2014 年 8 月 8 日時点で、MIT Saliency benchmark の総合評価値が 1 位であった。以下に詳細を記す。

モデル CovSal, BMS, Judd の出力である顕著度マップをそれぞれ S_{CovSal} , S_{BMS} , S_{Judd} と表すこととする。また線形重みをそれぞれ w_{CovSal} , w_{BMS} , w_{Judd} と表すこととする (ただし $w_{\text{CovSal}} + w_{\text{BMS}} + w_{\text{Judd}} = 1$)。新規注視モデルの顕著度マップ S_{mix} は

$$S_{\text{mix}} = w_{\text{CovSal}} \cdot S_{\text{CovSal}} + w_{\text{BMS}} \cdot S_{\text{BMS}} + w_{\text{Judd}} \cdot S_{\text{Judd}} \quad (1)$$

0)

で定式化される.

最も精度の高い線形重みを求めるには, その性能を定量的に評価するために基準が必要となる. MIT benchmark site では Saliency モデルの情報が記載されており, これらはいくつかの評価基準を用いて評価されている. MIT benchmark site が開設された 2012 年 1 月から 2014 年 8 月 12 日まで, モデルの定量的評価方法は (i) 信号検出理論に基づく AUR (Area Under ROC curve), (ii) 2 種のマップを 2 次元関数とみなして類似性を評価する Similarity, (iii) 2 種のマップを 2 つの画像とみなしその類似度を測る EMD (Earth Mover's Distance: 画像検索に用いられる距離尺度) が採用されていた. AUR と Similarity は数値が高いほど精度が高く, EMD は数値が低いほど精度が高いことを意味している. 本節で作成するモデルはこれら 3 種の評価方法に対して優れた性能を持つものとする. そのための総合評価量として以下の TOTAL を設定した.

$$\text{TOTAL} = \frac{\text{AUR} \cdot \text{Similarity}}{\text{EMD}} \quad (11)$$

新規注視モデルはこの TOTAL の値が最大値をとるものとする.

TOTAL の値を最大にする最適な重み値 $\vec{w} = \{w_{\text{CovSal}}, w_{\text{BMS}}, w_{\text{Judd}}\}$ を決めるために, 被験者に提示した 1003 枚の画像とこれらに対応する 1003 枚の注視マップを用いた. 使用した画像は MIT data set として公開されている²². この結果を Fig. 25 に示す. これは各モデルに対する重み \vec{w} に対する新規注視モデルの顕著度マップの評価値を色の濃淡

²² <http://people.csail.mit.edu/tjudd/WherePeopleLook/index.html>

で示したものである。評価値TOTALは重みを $\vec{w} = \{0.2362, 0.3161, 0.4477\}$ とするとき最大となることがわかった。この時の重みをもつ新規モデルを Mixture of Saliency Models (MSM) と名付けた。MSM の総合評価値はいずれのモデルよりも高い値を持つ。画像によっては改悪されるものもあったが、1003 枚全体では統計的に有意な改善であった(2 項検定; $p < 10^{-6}$)。このことから既存モデルを線形結合する単純なモデル構築手法であっても、有効性の高い新規モデルを開発できる可能性があることを示唆する。しかし、MSM の構築に用いた 1003 種に対して過度に適合(オーバーフィッティング)しており、他のデータでは性能が下がる可能性がある。そこで MSM の汎化性能を評価するために、注視マップを公開していない画像 300 枚を用いた性能評価を MIT benchmark チームに依頼した。その結果、2014 年 8 月 8 日時点で 1 位の総合評価値が得られた。なお MSM が 1 位となった 4 日後に評価方法が変更されたり、新たに SALICON[51] と呼ばれる高い評価値を示すモデルが提案されたりしたため、2016 年時点で MSM は 1 位ではない。SALICON の MSM への追加や、Judd モデルと置換した場合に新たに 1 位を獲得する可能性があるが、このモデルの評価については今後の課題とする。

5.2.4. 結果の考察

同種モデルを線形結合させた MSM の評価値が、高い値を持つ理由について考察する。

Fig. 23 左上を入力とした場合、コンポーネントとして使用した、既存モデル CovSal, Judd, BMS の出力を Fig. 23 下段に示す。Fig. 23 の上段中央は grand truth であるヒトの

注視マップを, Fig. 23 右上は MSM モデルの顕著度マップを示す. 注視マップを見ると, ヒトは山の頂上と峰, 海岸沿いの建物に対して注視位置を集中させている.

Fig. 23 下段の画像からもわかるように, 合成に使用した 3 種類の注視モデルはそれぞれ異なる位置を注視している. BMS は山頂と建造物に対して位置推定できているが峰の位置の推定には失敗している. CovSal は峰の推定はできていることから, それぞれのモデルには一長一短があることがわかる. 各モデルは異なる計算原理や視覚特徴を基に構築されており, それが結果の違いとして表れていると考えられる. MSM は各モデルが用いている視覚特徴の一長一短を吸収し, 他のモデルよりも高い評価値を示している. 実際に, 他の 1002 枚の画像に対する注視マップと顕著度マップを目視で比較すると, 同様の傾向が多く見られた.

なお複数のモデル結合による性能の向上は, AdaBoost 周辺の研究成果[52]から予測できる結果ではある. しかしながら実際に注視モデルを結合する試みは, MIT benchmark 実施時の 2014 年 8 月までには存在しなかった. このことは, モデルを結合できるプラットフォームが存在していることの有効性と優位性を示している.

5.3. 既存速度選択性モデルの改良

脳研究では日々新たな知見が発見され, それを再現する数理モデルが多く提案されている. 実際のモデル研究ではモデルを新規開発することが多いため, 既存モデルとの差異が明確ではない. また既存モデルと共通する箇所も新規にプログラミングされており,

モデル構築時に「車輪の再発明」を行っているのが現状である。しかし、再利用しようとする場合、既存モデルのプログラムは可読性についてあまり考慮していないため、それ自体が難しいという問題もまた存在する。

HI-brain ではモデルのコードの再利用が容易であることを示すために、既存モデルのコードを再利用したモデル改良を行った。MT 野の速度選択性を再現したモデルとして一般的である Simoncelli & Heeger モデル(SH モデル)[8]を既存モデルの例として使用し、2.3 節で紹介した Nishimoto らの最新知見[15]を基に SH モデルに改良を加えた。具体的には Simoncelli らが公開している MATLAB コードを複数の構成要素に分け、OpenRTM-aist のコンポーネントとして再実装、必要最低限のコンポーネントに対して修正を加えることで既存モデルとの差異を明確にした。まずは Simoncelli が提案するモデルについて説明する。

SH モデルは MT 野の速度選択性を再現した代表的なモデルである。MT 野内のニューロンは、その受容野に与えられた刺激が「特定の方向」に「特定の速度」で動いている場合に強く発火する性質を持つ[53]。同様に、初期視覚野の一つである V1 野にはある速度に対して選択的なニューロンが存在する。V1 野の速度選択性は、「特定の方向」に「特定の速度」で動く刺激に対して選択的であるが、さらにその刺激が「特定の時空間周波数帯」である場合のみ強く発火する。これら領野は独立したものでなく、V1 野から MT 野への情報の伝達経路が存在することが知られている[54]。SH モデルは V1 野

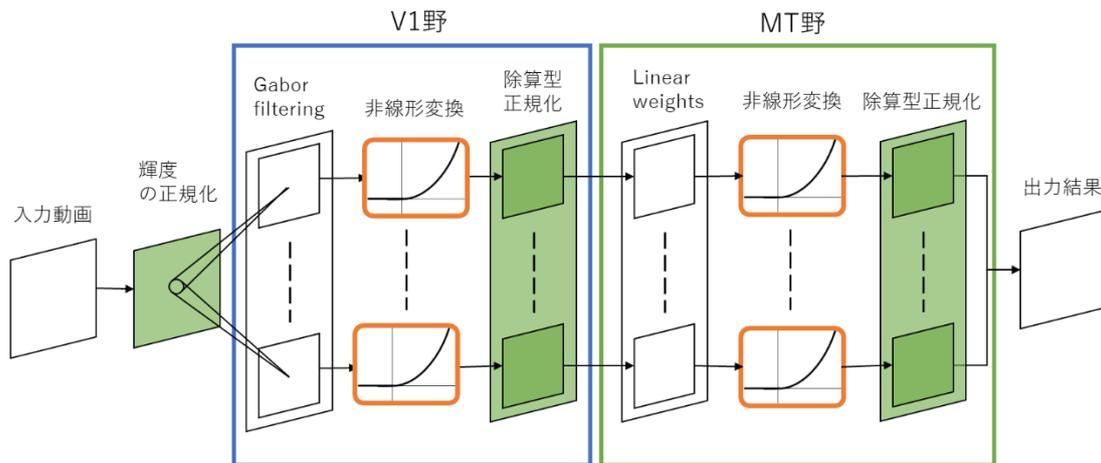


Fig. 26: Simoncelli & Heeger モデルの構成図. V1 野と MT 野の 2 段階で構成される.

V1 野と MT 野はともに同一の動きを検出するが, V1 野の速度選択性は時空間周波数に依存する. 同一の動きに選択的な V1 ニューロンの出力を組み合わせることで, MT 野の時空間周波数に依存しない速度選択性を再現する.

と MT 野の関連性と, MT 野の速度選択性を再現している. 故に SH モデルは V1 野と MT 野の 2 段階の構成によって構築され, MT 野の速度選択性を再現している.

SH モデルは, V1 野と MT 野の 2 つの段階で構成されている(Fig. 26). 初めに, V1 野の前段階の処理として, 与えられたパターンの正規化を行う. これは網膜上で行われている処理と近似しており, 脳領野に入力を与える前段階の処理として重要である[55]. 正規化された入力が V1 野に与えられる. V1 野は様々なガボールフィルタでのフィルタリングと非線形変換, 除算型の正規化によって構成されており, 数種類の V1 ニューロンを再現している. MT 野では線形和, 非線形変換, 除算型正規化によって構成され, 速度選択性ニューロンを再現する.

時空間周波数と速度の関係は以下の式で表される：

$$f_t = f_s V \quad (11)$$

速度(V)は空間周波数(f_s)と時間周波数(f_t)の比で表される。空間を 2 次元として考えた場合、 f_s は任意の方向の周波数となる。これを水平方向の周波数 f_x と垂直方向 f_y で考えた場合、速度 V は以下の式で表される：

$$f_t = (f_x \cos \theta + f_y \sin \theta) V \quad (12)$$

θ はその MT の選択的運動方向を意味する。ゆえに一つの速度は、2 次元フーリエ空間内では一つの線、3 次元フーリエ空間内では一つの面として表される(Fig. 27-(A)).

SH モデルではこの速度を検出するために、この面に沿うように興奮性の受容野をリング状に配置し、抑制性の受容野がこれの周囲に分布した形をしている(Fig. 27-(B)). これは時空間周波数(f_t, f_x, f_y)のすべての組み合わせで刺激を作成し、それを SH モデルの入力として与え、その出力結果を 3 次元空間上に分布させたものである。ある値を閾値とし、閾値以上の値を出力とする周波数帯を赤で表している。また赤の濃淡によって反応の強度を表す。

しかし実際の大抵の MT 野ニューロンの興奮性受容野の形状は、リング状の受容野だけでなく様々な形状を持つ(Fig. 28). Nishimoto らはより現実に即した MT の反応を見るために、一般的に実験で一般的に使用されているグレーティングなどの合成刺激でなく、自然動画に近いものを入力として使用した。この結果、実際の MT の受容野はリングの一部(f_t が 0 に近い部分)が欠けた形状の受容野(分離リング型受容野)、または f_t が 0 に近

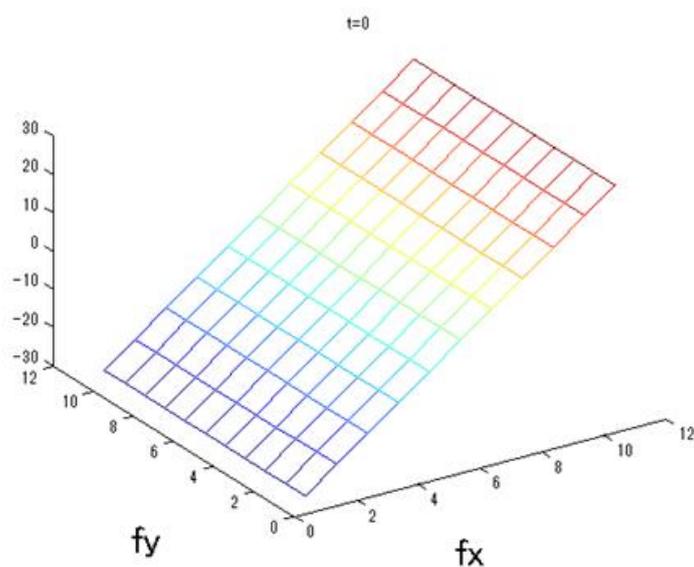
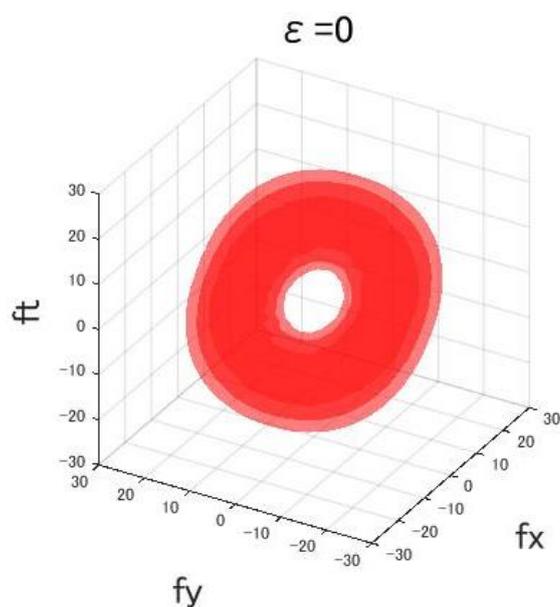
(1) 速度面($V = 1, \theta = 0$)(B) SH モデルの受容野($V = 1, \theta = 0$)

Fig. 27:速度と時空間周波数の関係. $V = 1, \theta = 0$ の場合は $f_t = f_x$ となるため、傾き 1 をもつ平面となる. (B)は赤い領域に該当する時空間周波数を持つ速度に対して選択的であることを意味する. この領域は(A)の速度面と重なる.

い領域では反応の強度が弱いため、分離リング型受容野を薄く結合させたような形状を

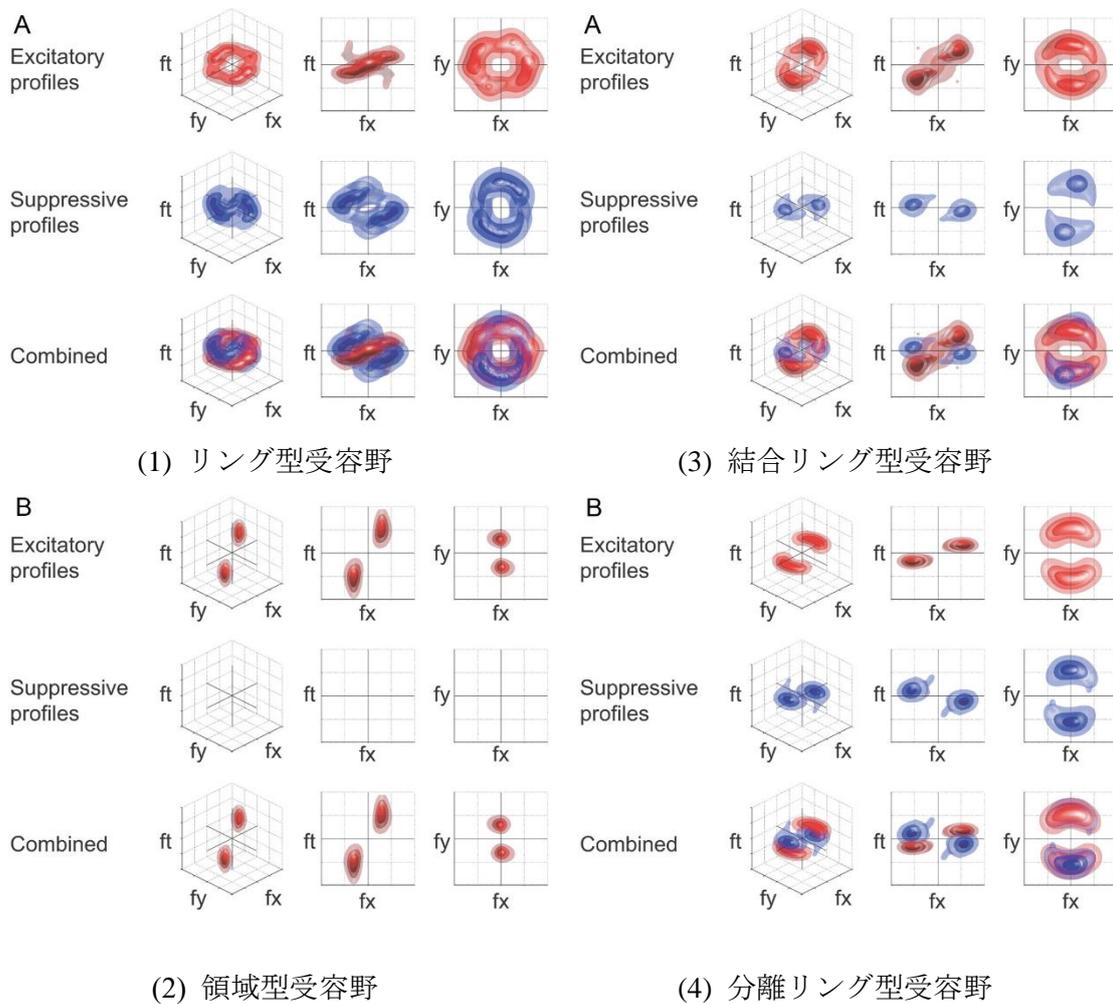


Fig. 28: Nishimoto らによって観測された MT 受容野の 4 種類の形状. SH モデルでは (1)のリング型受容野のみ再現している. 実際の MT 受容野は(3)(4)の形状を持つものが多く, (1)(2)は MT ニューロンのごく一部のみである.

持つもの(結合リング型受容野)の 2 種類の形状を持つものが MT 受容野の大半を占めることが判明した(Fig. 28-3,4).

対照的に, SH モデルのような完全なリング状の受容野(リング型受容野)や, 速度面に対して選択的ではなく周波数空間の一部に対して選択性を持つ受容野(領域型受容野)はごく少数であった(Fig. 28-1,2). ゆえに SH モデルは MT 野ニューロンをすべて再現し

ているのではなく、様々な形状を持つ MT 受容野の中のごく一部を再現している。

3.4 節では、Nishimoto らはこれら MT ニューロンを再現する定量的モデルを構築するために、モデルフレームワークの内部構造を変化させ最も精度の高いモデル構造を明らかにした。ここではゼロから新規モデルを構築するのではなく、既存の SH モデルのプログラムに対して改良を加えることで、4 種類の MT 受容野を再現するモデルを構築する。

SH モデルを改良するために、実際の MT ニューロンの多くは f_t が 0 の場合は反応強度が弱まる点に着目した。SH モデルが再現するリング型受容野は、周波数に依存しないある一つの速度を検出する。 $\theta = 0, V = 1$ の場合は、 $f_t = f_x$ に該当する周波数を持つ動きを検出する。しかし、 $f_t = f_x = 0$ の場合もこれに含まれる。つまり動きの見られない刺激に対してもこの受容野は動きを検出する。例えば $\theta = 0, V = 1$ の MT 受容野は、静止した水平方向の縞に対して動きを知覚する。それに対して分離リング型受容野は、 f_t, f_x がゼロに近い場合はこれを動きとして知覚しない。実際に人は $f_t = 0$ の動きに対して、動いていると知覚することは難しい。この $f_t = 0$ の刺激に対する反応強度の違いが MT 受容野の形状を変化させていると考えることができる。

作成するモデルでは刺激に含まれる MT の選択的運動方向の時空間周波数が小さい場合は反応強度を抑制する。その抑制の範囲及び抑制の強弱によって受容野の形状が変化する。例えば、反応強度の抑制が一切ない場合はリング型受容野を形成する。時空間

周波数が低い領域に強い抑制を与えた場合、分離リング型受容野となる。この抑制を弱めることで結合リング型受容野となる。そして、抑制が強く、範囲が広い場合は領域型受容野を形成する。この抑制の範囲と強度を決めるパラメータ ε, δ を新たに定義し、作成するモデルが検出する速度を以下のように定義する：

$$\begin{aligned} V'(f_x, f_y, f_t, \theta; \varepsilon) &= \frac{f_s^2}{f_s^2 + \varepsilon^2} V \\ &= \frac{(f_x \cos \theta + f_y \sin \theta)^2}{(f_x \cos \theta + f_y \sin \theta)^2 + \varepsilon^2} \frac{f_t}{f_s} \end{aligned} \quad (13)$$

V' は作成するモデルが検出する速度である。理想的な速度である V に対して、重みを与えている。 ε によって空間周波数が小さい場合にその反応強度を抑える効果がある (Fig. 29-1)。リング型、分離リング型、領域型受容野を比較した場合、これらの差異は抑制範囲の違いが原因であると考えられる(ε が0の場合はリング型となる。値が大きくなるに従い、分離リング型、領域型受容野へと変化する)。

次に抑制の強度を決めるパラメータ δ を導入した。その式が以下のとおりである。

$$\begin{aligned} V'(f_x, f_y, f_t, \theta; \varepsilon, \delta) &= \frac{f_s^2 + \varepsilon^2(1 - \delta)}{f_s^2 + \varepsilon^2} V \\ &= \frac{(f_x \cos \theta + f_y \sin \theta)^2 + \varepsilon^2(1 - \delta)}{(f_x \cos \theta + f_y \sin \theta)^2 + \varepsilon^2} \frac{f_t}{f_s} \end{aligned} \quad (14)$$

変数 ε の導入によって、空間周波数が低い領域に対する反応強度を抑える。しかしこれだけでは、結合リング型受容野を再現することは難しい。 δ の値が1に近いほど抑制強度は低下し、 δ が0に近いほど抑制強度は増加する (Fig. 29-2)。

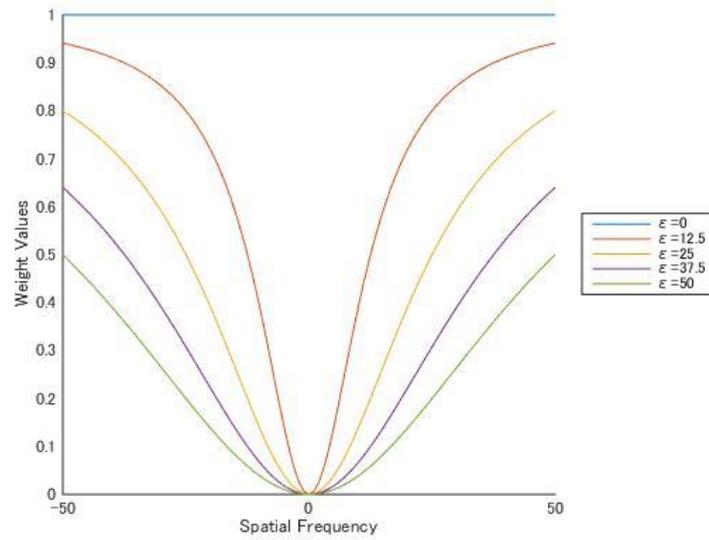
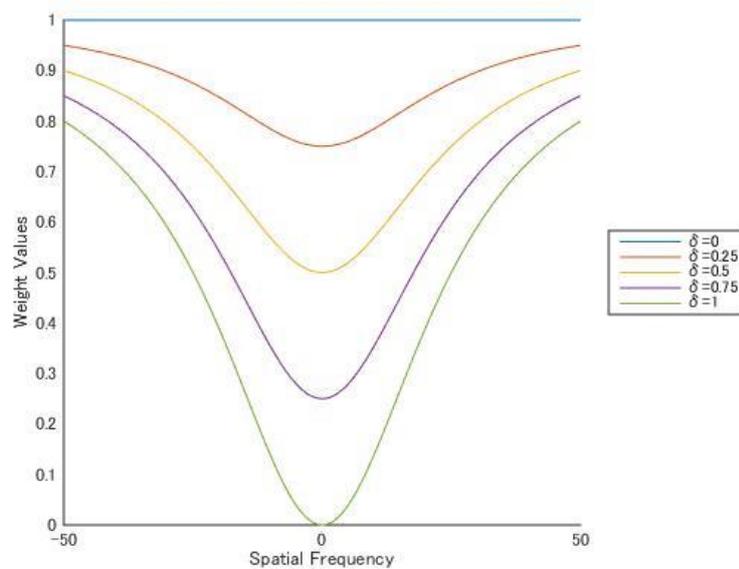
(1) ϵ による変化(2) δ による変化

Fig. 29: パラメータによる重みの変化.

これらパラメータを適切に設定することで、様々な MT ニューロンの受容野を再現することが可能である (Fig. 30). (1)は Fig. 28-1 と同様のリング型受容野を再現し、(2)は

Fig. 28-2 と同様の領域型受容野を再現している。これらは ε^2 の値を極端に大きくまたは小さくすることで再現することができる。(3)は Fig. 28-3 と同様の 結合リング型受容野を、(4)は Fig. 28-4 と同様の分離リング型受容野を再現している。これらはそれぞれ異なる抑制強度を与えることによって再現することが可能である。

SH モデルのプログラミングコードを改良することで新規モデルを実装した。

Simoncelli らは MATLAB で記述された SH モデルを公開しており、ダウンロードし実行することで SH モデルのシミュレーションを行うことができる²³。このコード上では SH モデルは 9 つの関数で構成されている(Fig. 31-A)。V1 野はフィルタリング・全波整流・ブラーリング。正規化の 4 つの関数から、MT 野はフィルタリング・ブラーリング・半波整流・ブラーリング・正規化の 5 つの関数から構成されている。まずはこれら MATLAB 関数を HI-brain 環境のコンポーネントとして再構築、必要最低限の改良で新規モデルの構築を行う。その結果、MT 野のフィルタリングコンポーネントを改良することで、前述した ε と δ を導入することができた(Fig. 31-B)。

HI-brain 上での既存モデル改良例として、SH によって提案された MT モデルを、最新の知見に従って改良を加え、新たな MT モデルとして提案した。MT は様々な形状の受容野を持つが、SH モデルはリング型受容野のみ再現する。我々のモデルは多様な MT 受容野の形状の違いを 2 つのパラメータの違いで再現することを可能にした。これ

²³ <http://www.cns.nyu.edu/~lcv/MTmodel/>

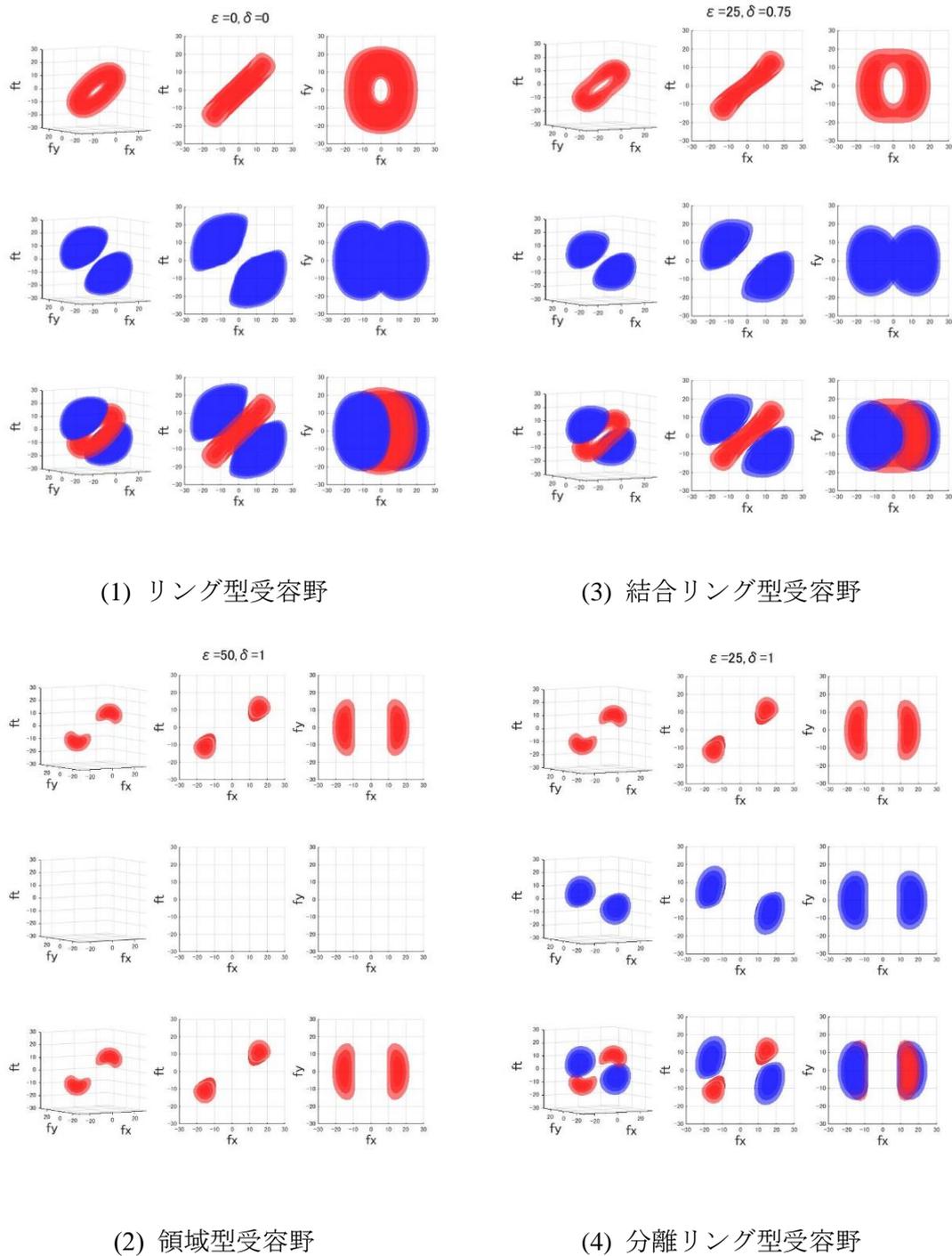
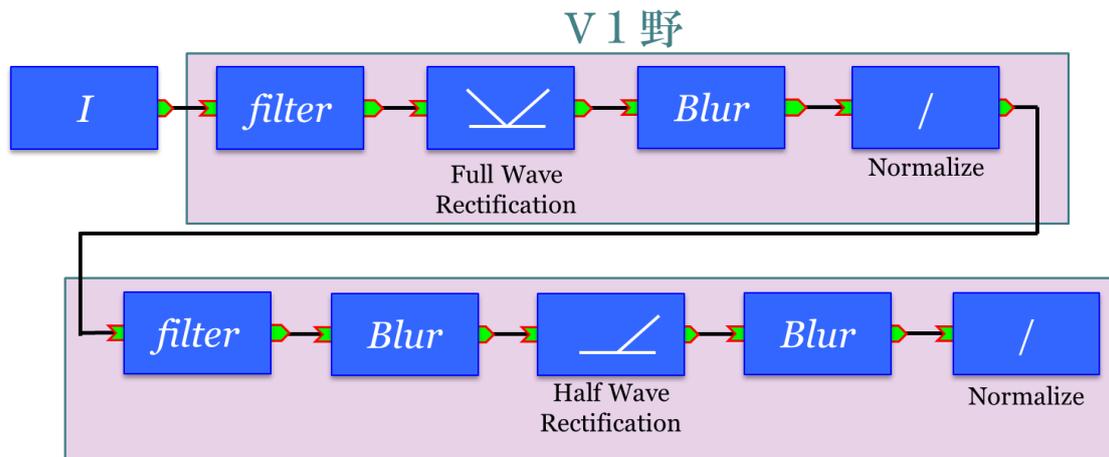
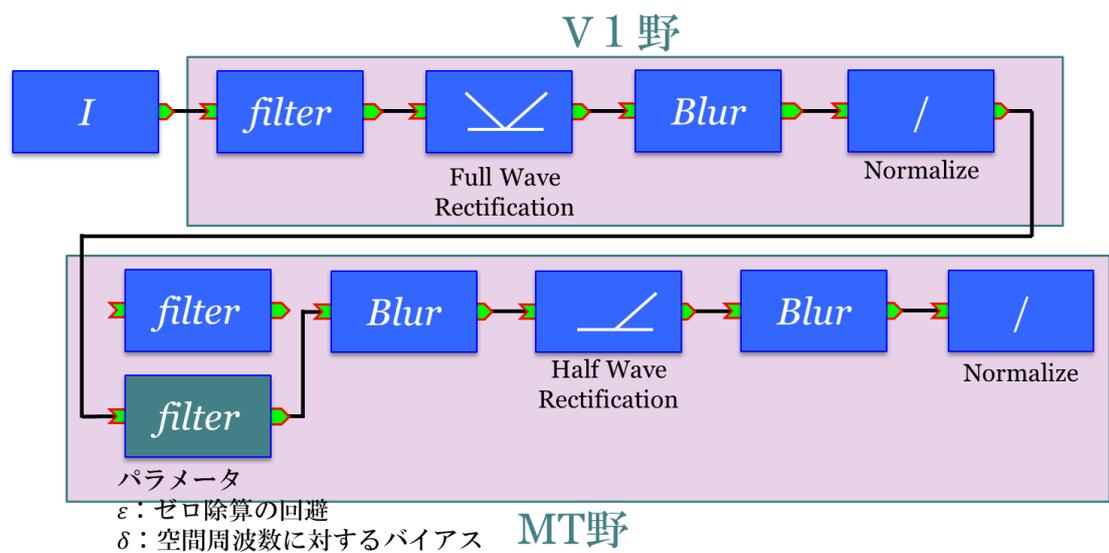


Fig. 30: シミュレーション結果から得られた MT 受容野.

らは低い時空間周波数に対する抑制の範囲と強度を決めるパラメータである. 抑制範



(A) コンポーネント化した SH モデル



(B) SH モデルを一部改良して作成した新規モデル

Fig. 31: コンポーネント化した SH モデルと改良した新規モデル.

囲の違いによって SH モデルのリング型受容野は様々な形状に変化し、リングの強度が変化する。これらパラメータの違いによって、様々な MT 受容野の形状を連続的に変

えることができる。故に受容野の形状の違いは低時空間周波数に対する信頼度として考えることができる。

5.4. 並列分散処理による大規模モデル実装例

本章では、HI-brain 環境上で様々なモデル構築が可能であることをいくつか例を挙げで示した。HI-brain の優れた点の一つに、様々な環境でのモデル実装が可能である点がある。本節では HI-brain の汎用性を示すために、複数台の RaspberryPi を使用した並列分散処理による視覚モデルシミュレーションを行う。一般的な計算機以外でもモデルの動作が可能であり、容易に並列分散処理が可能であることを示す。

Raspberry Pi はコンピュータ科学教育を目的とした小型計算機である。Raspberry Pi に対応した様々な OS が公開されており、microSD に OS をインストールすることで一般的な計算機と同様に操作可能である。一般的な計算機と比べて非常に安価であり、複数台のマシンを容易に揃えることができる。本節で使用する Raspberry Pi はその一種である Raspberry Pi 2 Model B²⁴を使用する。これは 900MHz の 4 コア CPU と 1GB の RAM を搭載している。また個々の Raspberry Pi の計算処理結果を表示するためのディスプレイとして、Adafruit PiTFT - 320x240 2.8" TFT+Touchscreen for Raspberry Pi²⁵を搭載させた。

実装するコンポーネントは、以下の 4 種類を用意した。

²⁴ <https://www.raspberrypi.org/products/raspberry-pi-2-model-b/>

²⁵ <https://www.adafruit.com/product/1601>

① 網膜イメージ

カメラから取得したフレーム画像またはあらかじめ用意した画像を、網膜上に
投射された画像として出力するコンポーネント

② 網膜モデル

6.1 節と同様に、神経節細胞の大細胞系 (magno) と小細胞系 (parvo) をシミュ
レーションするコンポーネント

③ V1 モデル

V1 野ニューロンの性質の一つである方位選択性をシミュレーションするコン
ポーネント

④ AIT モデル(顔認識モデル)

AIT 野のヒトの顔に対して選択的なニューロンをシミュレーションするコン
ポーネント. AIT 野には特定のオブジェクトに対して選択的なニューロンが存在
しており, ここではヒトの顔を対象とする.

これら 4 つのモデルコンポーネントを作成し, Raspberry Pi 上に構築した HI-brain 環
境で動作させる. これらコンポーネントを結合させた様子を Fig. 32 に示す. Fig. 32-(A)
は全体の結合関係である. 各ディスプレイにはその Raspberry Pi での処理結果を表示さ
せている. Raspberry Pi は LAN ケーブルを介して繋がっており, モデル間のデータ授受
は全て LAN ケーブルを介して行われる. Fig. 32-(B)(C)(D)(E)は個々のモデルの様子を見

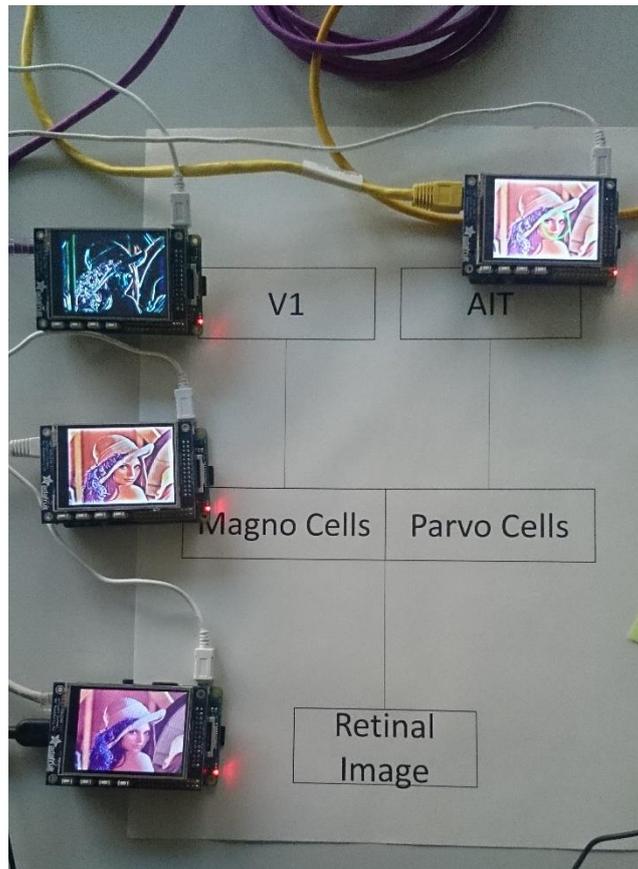
るために Fig. 32-(A)を拡大表示したものである。網膜イメージのコンポーネントは出力する画像を表示しており、この画像は網膜モデルに与えられる。網膜モデルでは大細胞系・小細胞系両方の処理を行っているが、ここでは小細胞系のシミュレーション結果を表示させている。Fig. 32-(B)と Fig. 32-(C)で表示している画像を比較すると、(C)の画像の細部が強調されていることがわかる。このシミュレーション結果は V1, AIT のモデルに与えられる。V1 野モデルでは垂直な方位に対して選択的なニューロンをシミュレーションしている(Fig. 32-(D))。AIT 野モデルでは画像内からヒトの顔を検出していることがわかる(Fig. 32-(E))。

モデルの実装例として、複数台の Raspberry Pi を利用したモデル実装を示した。各 Raspberry Pi 上に異なるモデルを実装することで、Raspberry Pi を物理コンポーネントとして構築した。それらがどのような処理が行っているかを理解しやすくするために、Raspberry Pi にディスプレイを搭載し表示させることで、シミュレーション結果が一目でわかるようになっている。

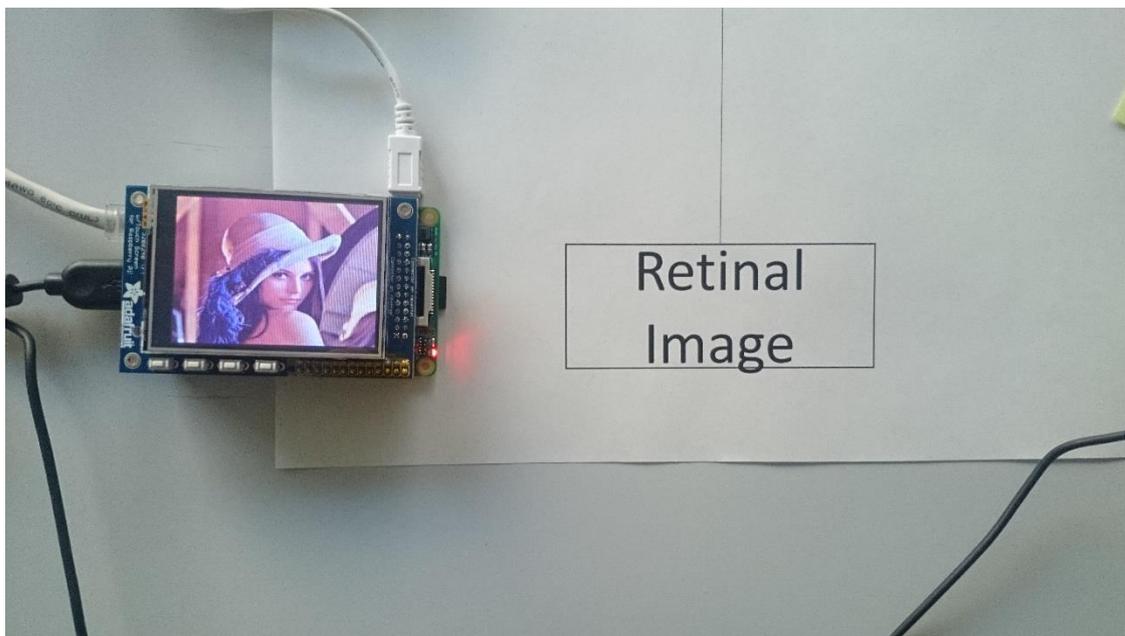
Raspberry Pi にモデルを構築する利点の一つに、モデルとその動作環境を SD カード内に格納することができる点がある。Raspberry Pi にはハードディスクではなくストレージはすべて SD カードで賄う。この SD カード内に OS をインストールして動作させているため、Raspberry Pi で動作させたいモデルを変更する場合は SD カードを入れ替えるだけでそれが可能となる。また他研究者にモデルを渡す場合でも、これまではそ

のモデルプログラムの動作環境をまずは用意する必要があったが、Raspberry Pi ごと相手に渡すだけでモデルの共有を行うことができる。

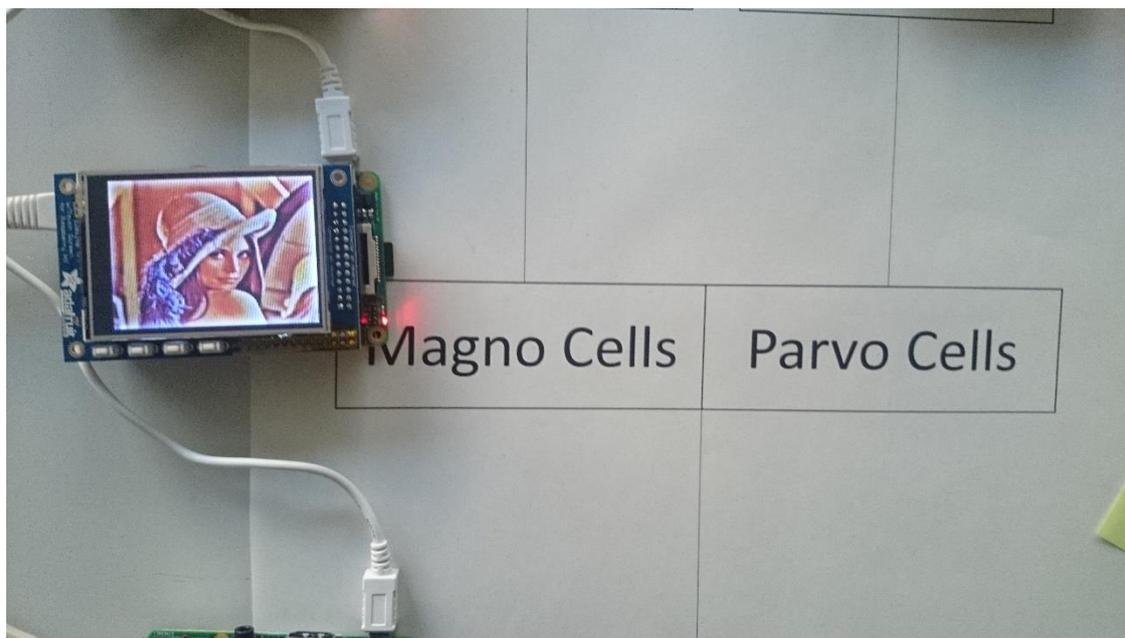
本節では2つの領野モデルと1つの器官モデルを結合させている。これらは視覚機能に関わる領野全体から見るとごく一部であるが、Raspberry Pi の台数と実装するモデルを用意することができれば、Raspberry Pi でも十分に大規模モデルのシミュレーションが可能であると考えられる。実際に Simon らは64台の Raspberry Pi を使用した計算機クラスタによる並列計算基盤を構築している[56]。また Raspberry Pi は組み込み機器としての需要があり、それは様々なロボットや電子機器に作成した視覚モデルをそのままロボットなどに実装可能であることを意味する。



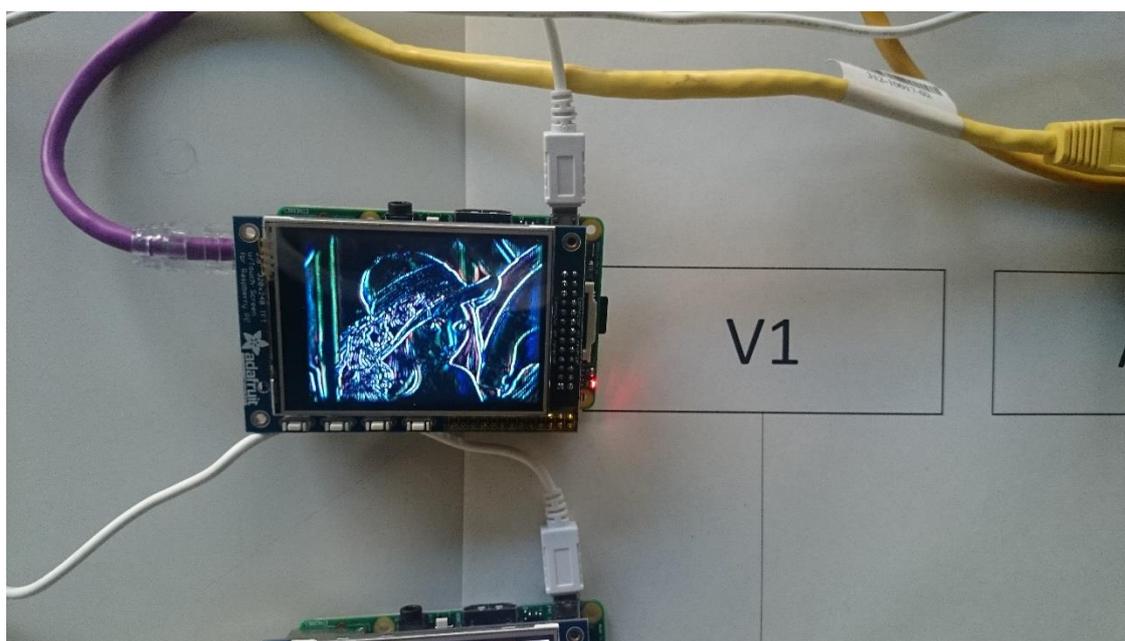
(A)



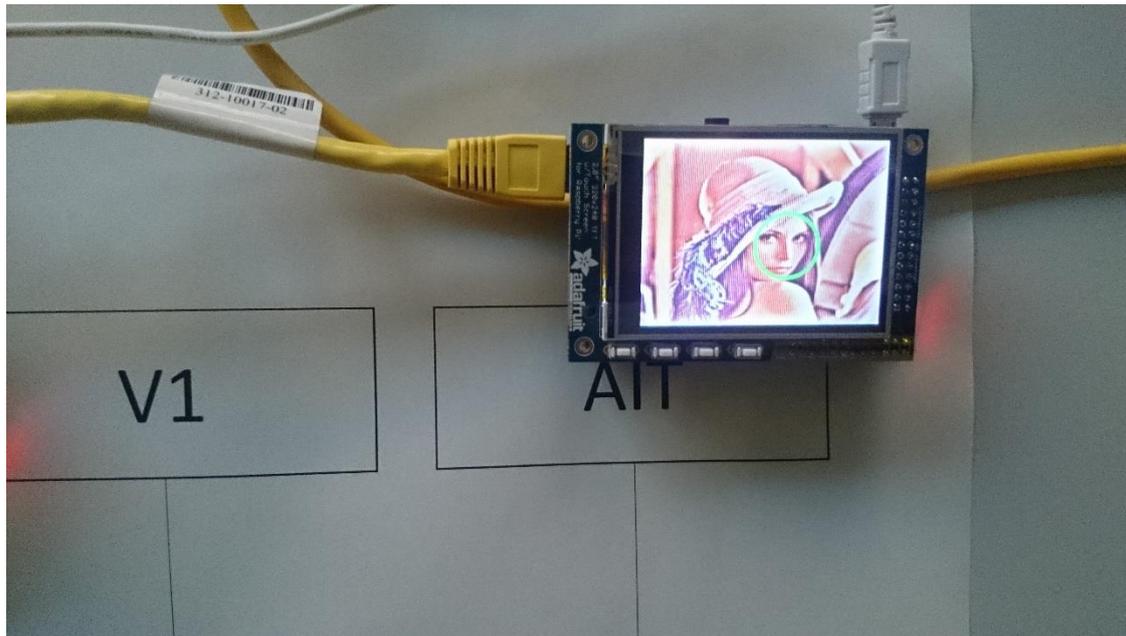
(B)



(C)



(D)



(E)

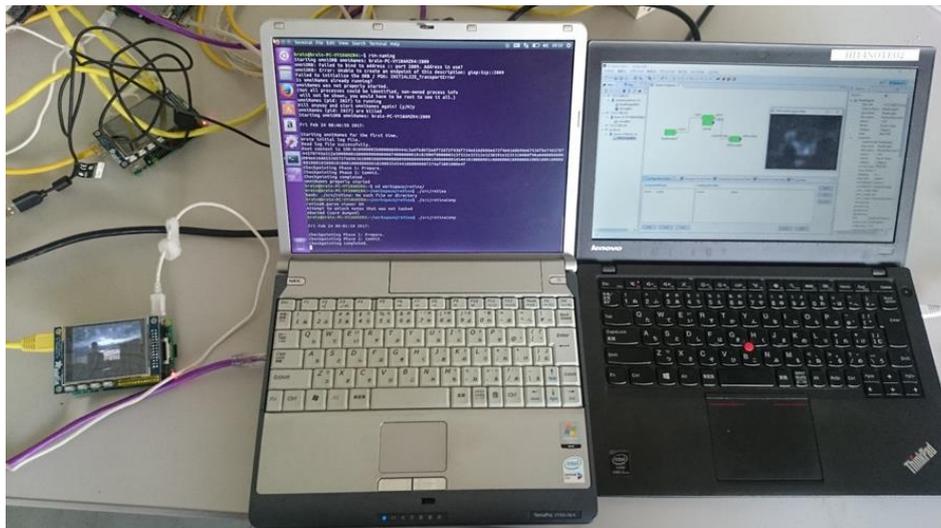
Fig. 32: Raspberry Pi でのモデル構築例. (A)は結合関係の全体像である. 個々のコンポーネントが異なるモデルを実装しており, 網膜イメージ, 網膜モデル, V1 モデル, AIT モデルから構成される. これらは LAN ケーブルを介してデータの授受を行う. (B)(C)(D)(E)はそれぞれのコンポーネントの様子を拡大した画像である.

5.5. 異なる環境で実装されたモデルの結合例

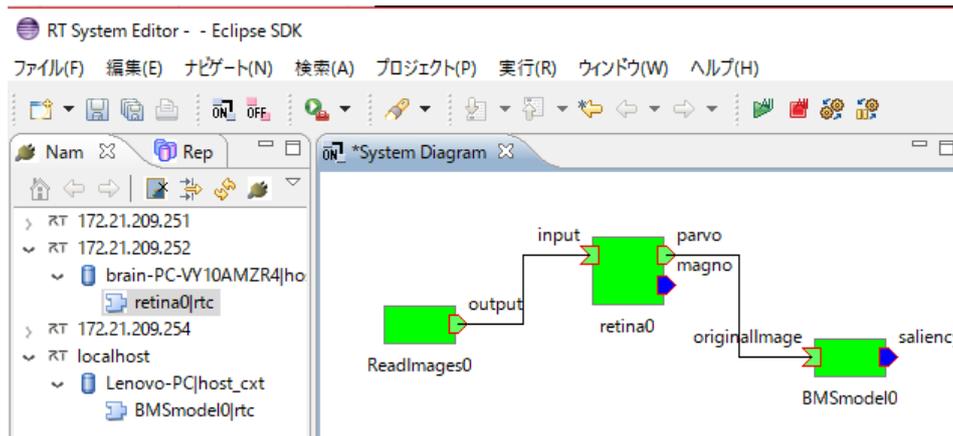
協調的モデル構築を行うにあたって問題となるのが構築環境の違いである. 視覚モデル研究において標準的なモデル構築環境は存在しないため, 各研究者は自ら得意とする環境下でモデルを作成する. 研究者が使用する環境は研究者自身が作成したモデルを動作させるのに十分ではある. そのためそれぞれのモデル研究においては何も問題はない.

しかし他の研究者が作成したモデルを動作させるには、そのための環境を用意しなければならない。必要なライブラリや無償ソフトウェアの導入には手間がかかり、さらに OS が異なる場合や有償のソフトウェアを必要とする場合はそれ以上の手間やコストを必要とする。また複数のモデルを結合させるには、同一環境上に動作環境の異なるモデルを協調的に動作させる必要がある。結合させるモデルの数に応じて手間は膨大となるため、協調的モデル研究がより困難となる。本研究で構築したソフトウェア基盤は、様々な環境上への導入、複数の計算機を使用した並列分散処理を可能とすることで実装環境の違いによるモデル結合の手間を解消する。

本節では単一の環境上に実装しなおすのではなく、異なる環境同士の結合によるモデル結合を本基盤が可能にすることを示す。本基盤は様々な OS(Windows, MacOS, Linux など)やプログラミング言語(C++, Java, Python など)に対応しており、様々なハードウェア及びソフトウェア環境でのモデル実装が可能である。本基盤はこれら異なる環境上のモデルの結合を可能とする。具体的には、入出力データを記述するデータフォーマットが一致していればモデル同士を結合させることができる。また LAN ケーブルを介したデータ通信によって、異なる計算機上のモデル同士のデータ授受が可能である。そのため本基盤が導入された様々な環境のモデルであっても協調的に動作させることが可能である。それを確認するために、一般的なノート PC2 台と、 RaspberryPi を使用したモデル結合を行った。Fig. 33 は実装した計算機を結合させた様子と、モデルの結合図を



(A)



(B)

Fig. 33: 異なる計算機や OS 上で実装されたモデルを結合させた様子. (A)は使用した計算機をモデルの結合関係と同様に並べたものである. (B)は実装させたモデルの結合関係を示す.

示す. Fig. 33(A)はモデル実装に使用した計算機を結合順に並べてものである. Linux ディストリビューションである Raspbian と Ubuntu, Microsoft 社の Windows10 をそれぞれ

れインストールしている。RaspberryPi には画像を出力するコンポーネントを、中央のノート PC 上には網膜モデルを、右のノート PC には Saliency モデルコンポーネントをそれぞれ実装している。Fig. 33(B)は OpenRTM-aist のツールの一つである RT System Editor が各ハードウェア上のモデルを認識している様子である。RT System Editor は 3 台のハードウェアのうち Windows10 をインストールした PC 上で動作させた。Fig. 33(A)の右端のディスプレイからもその様子がわかる。このように同一ネットワーク上のハードウェアに実装されたモデルであれば、同一計算機上に存在しているようにモデルの操作や結合関係の編集を行うことができる。

一般的にモデル同士を結合させる場合、単一のハードウェア及びソフトウェア環境でモデルを動作させ、入出力関係をプログラミングまたは手作業で制御する必要がある。モデル研究者の開発環境を限定させることで解決可能であるが現実的ではない。しかし本基盤を導入することで、様々な環境(ハードウェアや OS, モデルを記述するプログラミング言語)でモデル構築を行うことができ、またこれら環境同士のネットワーク結合によって、同一環境上のモデルのように結合させ動作させることができる。このことは、本基盤の利用者に対してハードウェアや OS を制限させずにモデル開発を行なわせ、モデル結合に必要な労力を削減できることを意味する。

6. まとめと考察

脳全体のモデル化を行うには、既存モデルの結合・置換・シミュレーションを可能とするソフトウェア基盤の存在が重要となる。しかし、既存の視覚モデルと既存のソフトウェア基盤についての調査の結果、機能レベル(疎粒度)のモデルの構築を目的としたソフトウェア基盤は存在していないことが分かった。本研究では、複数のモデルを組み合わせた大規模モデルを協調的に開発するためのソフトウェア基盤「HI-brain」の構築を目的とした。

まずは、既存の疎粒度モデルをいくつか挙げ、これらに共通したモデル構造に着目した。これらモデルが複数のコンポーネントによる階層構造を持つことから、本基盤に必要な要件とそれを満たすための必要機能を明らかにした。調査と考察の結果、必要機能をすべて満たす既存の脳研究用ソフトウェア基盤は存在していない。そこで本研究では、ロボット開発用の RT ミドルウェアと実装である OpenRTM-aist を基にモデル開発に必要な追加開発を行うことで、必要機能を満たしたソフトウェア基盤を構築した。

本基盤は視覚の疎粒度モデルを対象としているが、他の感覚野(聴覚や体性感覚など)などの脳モデルの構築も可能であると考えられる。OpenRTM-aist はロボット構築の実績が多数あり、ロボットアームなどの既存のハードウェアに対応したコンポーネントも公開されている²⁶。そのため、筋骨格系の数理モデルと視覚数理モデルの結合が可能で

²⁶ <http://www.openrtm.org/>

あり、それをロボットにそのまま実装することも期待できる。

実際のモデル構築例をいくつか示した。器官モデルと脳機能モデルの結合例として、網膜モデルと高次機能の一つである顔検出モデルとの結合を行い、脳神経系研究に対してコンポーネント指向でのモデル開発の重要性を示した。細胞単位でのモデル改良例として、運動視の一般的なモデルである Simoncelli と Heeger が提案したモデルを最新の知見に基づいて改良した。具体的には、まず Simoncelli らが作成したモデルのプログラムコードを9つのコンポーネントに分解した。HI-brainの主機能であるモデルの置換・修正によって、ある一つのコンポーネントを置換・修正した。その結果、新たな知見を反映させたモデルを構築することが可能であることを示した。複数の既存モデルを再利用したモデルの開発例として、複数の注視位置を予測するモデルを組み合わせ、より予測性能が高い新規モデルを効率よく作成し、定量評価値も既存モデルを超えることを示した。またモデルの実装例として、一般的な計算機だけでなく、様々な環境でモデル実装が可能であることを示すために、小型計算機である RaspberryPi 上にモデルを実装、LAN ケーブルを介した結合を行った。モデルコンポーネントをハードウェアレベルで構築することにより、モデル構築の並列分散処理の様子が一目で理解できる。また RaspberryPi とモデルが一对一对応しているため、他者へのモデルの譲渡が容易となる。さらに、異なる環境で構築されたモデルの結合が可能であることも示した。

本基盤は視覚モデルの協調的モデル研究に必要な要件を満たしている。しかし、

脳研究者に本基盤を利用するためには利便性が不足しているという問題点もまた明らかになった。一つは本基盤を利用するための作業手順や、使用方法が複雑であり、MATLAB などの商用ソフトウェアと比較すると脳研究者を対象としたものとしては使用が困難である点である。たとえば Windows7 (64bit) 上で本プラットフォーム環境を構築するには JavaJRE, VisualStudio, cmake をはじめ、その他 7 種類のソフトウェアの適切な事前設定とインストールが必要であった。OpenRTM-aist と同様の問題ではあるが、HI-brain の場合は boost や OpenCV ライブラリのインストールも必要とするため、手順はより複雑化している。これを解決するために、HI-brain 環境を標準でインストールした OS を用意した。しかし脳研究者にこれを実装するための物理または仮想環境のセットアップを期待しなければならない。また Linux は脳研究者にとっては一般的な OS ではない。脳研究者が通常使用する Windows や MacOS 向けの必要ソフトウェアと一括インストールなどの手段が必要となるだろう。

また HI-brain 環境は単一のソフトウェアでなく複数のソフトウェアの組み合わせによって構成されている。そのためインストールした個々のソフトウェアの役割や使用方法を熟知し、これらを適切に切り替えて利用しなければならない。MATLAB でのモデル開発では、モデルの作成や修正、シミュレーションなどは MATLAB の IDE での作業でほぼ完結している。しかし本基盤でモデルを結合させるには、まず Eclipse でコンポーネントの概要を記述し、テンプレートを作成する。次に cmake でビルド用ファイルの生

成, VisualStudio で実装およびビルドを行う必要がある(Windows の場合). そのため, これら困難さを軽減するためのソフトウェアとして, OpenCV-RTC を開発したが, Linux や MacOS などには対応しておらず, 使用方法を理解するためのドキュメントが必要となる.

モデル研究では主に MATLAB が広く利用されている. そのため代替となる本基盤は, MATLAB 同様の利便性が要求される. しかし, 実行中のコンポーネントとそのプログラムの所在情報が, ユーザの記憶やメモに依存しているなどの問題がある. Windows の場合, コンポーネントの結合関係の編集や実行は Eclipse で, コンポーネントのプログラムの修正は VisualStudio で行う必要がある. そのため実行中のコンポーネントを修正する場合は, Eclipse でコンポーネントの実行を停止させ, VisualStudio で修正する必要がある. 例えば, 実行中のコンポーネントをクリックすることで対応するソースコードが表示され, 修正もまた自動的に反映させるなどといった仕組みを用意する必要がある. つまり MATLAB や Simulink のような IDE の構築が必要である. これによって, ユーザの記憶やメモに頼る必要がないため, 脳研究者だけでなく OpenRTM-aist の主な利用者であるロボット研究者にとっても有益だろう. この問題は OpenRTM-aist の本質的な問題でなく, 追加開発を別途行うことで解決することができるため, 今後重要な問題として位置付けている.

協調的モデル研究を行うために, モデルデータベースの構築を行ったが, 現在の HI-

brain のホームページでは、単純なモデルの羅列のみで終わっている。今後は個々の数理モデルのメタデータ（脳領域や脳機能、依存関係など）の記述や、求められるデータベースの要件を十分に精査する必要がある。

これまでのモデル研究において、筆者の知る限り、過去のモデルを再利用することによって特筆すべき成果が得られたといった例はない。新しい計算原理に基づいた新規数理モデルの開発も重要である。しかし、これら既存研究の成果を基にした、発展的なモデル研究もまた重要になるのではないかとと思われる。

7. 参考文献

- [1] D. H. Hubel and M. S. Livingstone: Segregation of form, color and stereopsis in primate area 18, *Journal of Neuroscience*, 7, 3378-3415 (1987)
- [2] J. H. Maunsell and D. C. Van Essen: Functional properties of neurons in middle temporal visual area of the macaque monkey. II. Binocular interactions and sensitivity to binocular disparity, *Journal of Neurophysiology*, 49, 1148-1167 (1983)
- [3] M Taira, K. Tsutsui, M Jiang, K Yara and H Sakata: Parietal Neurons Represent Surface Orientation From the Gradient of Binocular Disparity, *Journal of Neurophysiology*, 83, 3140-3146 (2000)
- [4] K. Tanaka, K.: Neuronal mechanisms of object recognition. *Science* 262, 685-688 (1993)
- [5] A. Minami, K. Nakamura and K. Kubota: Neuronal responses to photographs in the superior temporal sulcus of the rhesus monkey, *Behavioral Brain Research*, 60, 1-13 (1994)
- [6] N.R. Carlson: *Physiology of behavior*, Pearson Education Ltd., UK (2013)
- [7] S. Marčelja: Mathematical description of the responses of simple cortical cells, *J. Opt. Soc. Am.*, 70, 1297-1300 (1980)
- [8] E.P. Simoncelli and D.J. Heeger: A model of neuronal responses in visual area MT, *Vision Research*, 38, 743-761 (1998)
- [9] Laurent Itti and Christof Koch: A saliency-based search mechanism for overt and covert shifts of visual attention, *Vision Research*, 40, 1489-1506 (2000)
- [10] A. Benoit, A. Caplier, B. Durette and J. Hérault: Using Human Visual System Modeling For Bio-Inspired Low Level Image Processing, *Computer Vision and Image Understanding* 114, 758-773 (2010)

-
- [11] N. Ando, T. Suehiro, K. Kitagaki, T. Kotoku and W. Yoon: RT-Middleware: Distributed Component Middleware for RT (Robot Technology), IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2005), 3555-3560 (2005)
- [12] A. Motoyoshi, S. Nishida, L. Sharan and E.H. Adelson: Image statistics and the perception of surface qualities, *Nature*, 447, 206-209 (2007)
- [13] Kunihiro Fukushima: Neocognitron: A Hierarchical Neural Network Capable of Visual Pattern Recognition, *Neural Networks*, vol.1, pp.119-130 (1988)
- [14] Laurent Itti and Christof Koch: Computational modelling of visual attention, *Nature Reviews Neuroscience* 2, 194-203 (2001)
- [15] Shinji Nishimoto and Jack L. Gallant: A Three-Dimensional Spatiotemporal Receptive Field Model Explains Responses of Area MT Neurons to Naturalistic Movies, *The Journal of Neuroscience*, 31(41) 14551-14564 (2011)
- [16] Kunihiro Fukushima: Neocognitron for handwritten digit recognition, *Neurocomputing* 51, 161-180 (2003)
- [17] Anitha Pasupathy and Charles E. Connor: Responses to contour features in macaque area v4, *J. Neurophysiol.* 82, 2490–2502 (1999)
- [18] Jochen Braun: Shape-from-shading is independent of visual attention and may be a ‘texton’. *Spatial Vision*, vol.7, 311–322 (1993)
- [19] Jennifer Sun and Pietro Perona: Early computation of shape and reflectance in the visual system. *Nature* 379, 165–168 (1996)
- [20] Quoc V. Le, Marc’Aurelio Ranzato, Rajat Monga, Matthieu Devin, Kai Chen, Greg S. Corrado, Jeff Dean and Andrew Y. Ng: Building High-level Features Using Large Scale Unsupervised Learning, *International Conference on Machine Learning*, (2012)
- [21] Desimone, R., Albright, T., Gross, C., and Bruce, C.: Stimulus-selective properties of

- inferior temporal neurons in the macaque. *The Journal of Neuroscience* (1984)
- [22] Hinton, G. E., Osindero, S., and Teh, Y. W.: A fast learning algorithm for deep belief nets, *Neural Computation* (2006)
- [23] Hinton, G. E. and Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks, *Science* (2006)
- [24] Bengio, Y., Lamblin, P., Popovici, D., and Larochelle, H.: Greedy layerwise training of deep networks, *NIPS* (2007)
- [25] Lee, H., Battle, A., Raina, R., and Ng, Andrew Y.: Efficient sparse coding algorithms, *NIPS* (2007)
- [26] Coates, A., Lee, H., and Ng, A. Y.: An analysis of singlelayer networks in unsupervised feature learning, *AISTATS*, 14 (2011)
- [27] Ciresan, D. C., Meier, U., Gambardella, L. M., and Schmidhuber, J.: Deep big simple neural nets excel on handwritten digit recognition, *CoRR* (2010)
- [28] Krizhevsky, A.: Learning multiple layers of features from tiny images, Technical report, University of Toronto (2009)
- [29] Le, Q. V., Ngiam, J., Chen, Z., Chia, D., Koh, P. W., and Ng, A. Y. Tiled convolutional neural networks. *NIPS*, 2010.
- [30] Raina, R., Madhavan, A., and Ng, A. Y. Large-scale deep unsupervised learning using graphics processors. *ICML*, 2009.
- [31] Le, Q.V., Ngiam, J., Coates, A., Lahiri, A., Prochnow, B., and Ng, A.Y. On optimization methods for deep learning. In *ICML*, 2011
- [32] Hyvärinen, A., Hurri, J., and Hoyer, P. O. *Natural Image Statistics*. Springer, 2009
- [33] Gregor, K. and LeCun, Y. Emergence of complex-like cells in a temporal product network with local receptive fields. *arXiv:1006.0448*, 2010

-
- [34] Jarrett, K., Kavukcuoglu, K., Ranzato, M.A., and LeCun, Y. What is the best multi-stage architecture for object recognition? In ICCV, 2009
- [35] Pinto, N., Cox, D. D., and DiCarlo, J. J. Why is real-world visual object recognition hard? PLoS Computational Biology, 2008
- [36] Lyu, S. and Simoncelli, E. P. Nonlinear image representation using divisive normalization. In CVPR, 2008.
- [37] Fukushima, K. and Miyake, S. Neocognitron: A new algorithm for pattern recognition tolerant of deformations and shifts in position. Pattern Recognition, 1982
- [38] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient based learning applied to document recognition. Proceeding of the IEEE, 1998.
- [39] Riesenhuber, M. and Poggio, T. Hierarchical models of object recognition in cortex. Nature Neuroscience, 1999.
- [40] Sermanet, P. and LeCun, Y. Traffic sign recognition with multiscale convolutional neural networks. In IJCNN, 2011.
- [41] Pakkenberg, B., P., D., Marner, L., Bundgaard, M. J., Gundersen, H. J. G., Nyengaard, J. R., and Regeur, L. Aging and the human neocortex. Experimental Gerontology, 2003.
- [42] S. Usui, K. Inagaki, T. Kannon, Y. Kamiyama, S. Satoh, N.L. Kamiji, Y. Hirata, A. Ishihara and H. Shouno: A Next Generation Modeling Environment PLATO: Platform for Collaborative Brain System Modeling, Neural Information Processing, LNCS5863, 84-90 (2009)
- [43] N. Ando, T. Suehiro and T. Kotoku: A Software Platform for Component Based RT-System Development: OpenRTM-Aist, International Conference on Simulation Modeling and Programming for Autonomous Robots, LNCS5325, 87-98 (2008)
- [44] Object Management Group, Robotic Technology Component Specification Version 1.0,

- formal/2008-04-04 (2008)
- [45] N. Ando, T. Suehiro, K. Kitagaki, T. Kotoku, and W. Yoon: Composite Component Framework for RT-Middleware (Robot Technology Middleware), 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM2005), 1330-1335 (2005)
- [46] R.L. De Valois and K.K. De Valois: Spatial Vision, p55, Oxford University Press, USA (1988)
- [47] S.M. Smirnakis, M.J. Berry, D.K. Warland, W. Bialek and M Meister: Adaptation of retinal processing to image contrast and spatial scale, *Nature*, 386, 69-73 (1997)
- [48] E. Erdem and A. Erdem: Visual saliency estimation by nonlinearly integrating features using region covariances. *J. Vis.*, vol. 13, 11 (2013)
- [49] J. Zhang and S. Sclaroff: Saliency Detection: A Boolean Map Approach, *ICCV*, 153–160 (2013)
- [50] T. Judd, K. Ehinger, F. Durand and A. Torralba: Learning to predict where humans look, *ICCV*, 2106–2113 (2009)
- [51] M. Jiang, S. Huang, J. Duan and Q. Zhao: SALICON: Saliency in Context, *CVPR*, 1072-1080 (2015)
- [52] D. Cireşan, U. Meier and J. Schmidhuber: Multi-column deep neural networks for image classification, *CVPR*, 3642-3649 (2012)
- [53] J.A.Movshon, E.H.Adelson, M.S.Gizzi, and W.T.Newsone: The analysis of moving visual patterns, *Experimental Brain Research Supplementum 11:Pattern Recognition Mechanisms*, pp117-151 (1986)
- [54] R. Dubner, S.M. Zeki: Response properties and receptive fields of cells in an anatomically defined region of the superior temporal sulcus in the monkey, *Brain*

-
- Research, vol.35, 2, 5289-532 (1971)
- [55] Shapley R, Enroth-Cugell C: Visual adaptation and retinal gain controls, Prog Retin Res, Vol.3, 263–346 (1984)
- [56] Simon J. Cox, James T. Cox, Richard P. Boardman, Steven J. Hohnston, Mark Scott, Neil S. O’Brien: Iridis-pi: a low-cost, compact demonstration cluster, Cluster Computing, vol.17, 2, 349-358 (2014)

関連論文の印刷公表の方法及び時期

論文誌

占部 一輝, 中村 大樹, 佐藤 俊治, 韓 雪花, “脳の数理モデル構築を目的とした RT ミドルウェアの応用と結果”, 計測自動制御学会論文集 52 巻 5 号

国際会議

Keiichiro Inagaki, Takayuki Kannon, Yoshimi Kamiyama, Shunji Satoh, Nilton Kamiji, Daiki Sone, Kazuki Urabe and Shiro Usui, "A large-scale whole visual system model integrated by PLATO and its implementation on high performance computer,"4th INCF Congress of Neuroinformatics, P043, Sep,2011

Kazuki Urabe, Shunji Satoh, Taihei Kitagawa, "Collaborative software platform for computational brain research based on OpenRTM,"SCIS ISIS 2012, W2-42-4, Nov,2012

Xuehua HAN, Shunji SATOH, Daiki NAKAMURA, Kazuki URABE, “Unifying computational models for visual attention yields better scores than state-of-the-art models”, Advances in Neuroinformatics 2014, Sep, 2014

国内学会

皆川保裕, 占部一輝, 佐藤俊治, 知久健, 川口仁, 長瀬雅之, “RT ミドルウェアをベースとした視覚研究用プラットフォームの開発”, SI2013,2013 年 12 月

中村大樹, 佐藤俊治, 韓雪花, 占部一輝, “視覚脳科学研究を目的とした RT ミドルウェアの応用と結果”, SI2014, 2014 年 12 月

韓雪花, 佐藤俊治, 中村大樹, 占部一輝, “Neuroinformatics 的観点から構築され

た新規 Saliency モデル”, 日本視覚学会 2015 年冬季大会, 2015 年 1 月

北川大平, 佐藤俊治, 占部一輝, “視覚数理モデル構築のためのプラットフォーム開発,” 日本視覚学会 2013 年冬季大会, 2013 年 1 月

Urabe K, Minakawa Y, Satoh S, Chiku T, Kawaguchi S, Nagase M, Yamaguchi Y,
“Integrated development platform for computational brain research: HI-brain”, 脳と心のメカニズム 第 14 回冬のワークショップ, 2014

占部一輝, 佐藤俊治, 中村大樹, “視覚研究用シミュレーション基盤: 数理モデルの結合, 追加及び置換を行うための手法”, 第 25 回 日本神経回路学会 全国大会, 2015 年 9 月

受賞

2014 年 12 月, RT ミドルウェアコンテスト 2014, 日本ロボット工業会賞

2014 年 12 月, RT ミドルウェアコンテスト 2014, ベストサポート賞受賞

2015 年 12 月, 第 1 回平成 27 年度ロボットビジネス推進協議会, RT ミドルウェア普及貢献賞

謝辞

本研究は長期の研究期間にわたり、多くの方々からのご助言やご指導を頂きました。特に研究の全過程を通してご指導、ご鞭撻を賜りました佐藤俊治准教授ならびに阪口豊教授には心より深謝いたします。

また中村大樹様、北川大平様、皆川保裕様、韓雪花様には本研究の共同研究者として協力していただきました。株式会社セックの長瀬雅之様、川口仁様、知久健様、川原健太郎様には共同開発者として多大な貢献をしていただきました。深く感謝申し上げます。

また博士課程の先輩である浅野哲理様には研究へのアドバイスだけでなく、学生生活でも修士からお世話になりました。

本研究は多くの方のご協力、ご指導によって完成することができました。本研究に関わっていただきました皆様、人間情報学講座の皆様には心から感謝し、お礼申し上げます。