# 脳活動に基づくプログラム理解の困難さ測定

中川 尊雄　亀井 靖高　上野 秀剛　門田 暁人　鵜林 尚靖
松本 健一

本論文は，NIRS (Near Infra-Red Spectroscopy; 近赤外分光法) による脳血流計測を用い，開発者がプログラム理解時に困難を感じているかの判別を試みた我々の先行研究（レター論文）を発展させたものである．本論文では，20名の被験者に対して，難易度の異なる三種類のプログラムの理解時の脳血流を計測する実験を行った．実験が中断された 3 名を除く 17 名中 16 名において，(1) 難易度の高いプログラムの理解時に脳活動がより活発化するという結果 (正確二項検定, $p < 0.01$) が得られた．また，(2) 被験者アンケートによって得られた難易度の主観的評価と，脳活動値の間には有意な相関 (スピアマンの順位相関係数 $= 0.46$, $p < 0.01$) がみられた．

This paper extends our previous study (letter paper), which quantifies the difficulty of program comprehension based on brain activation measured by NIRS (Near Infra-Red Spectroscopy) during source code reading. We performed controlled experiments with 20 subjects. 3 of 20 subjects could not complete the measurement. We found that: (1) 16 of 17 shows strong brain activation during reading of obfuscated program (binomial test, $p < 0.01$) and (2) subjective evaluation of difficulty is correlated with brain activation (Spearman's correlation coefficient $= 0.46$, $p < 0.01$).

## 1　はじめに

プログラム理解は，コーディング，テスト，保守など，ソフトウェア開発における幅広い工程で実施される重要な活動であり，成果物の品質に強く影響する．例えば，プログラムを理解できている人とそうでない人の間で，バグ発見効率に 10 倍以上の差があることが示されている [22]．しかし，ある人物がプログラム理解に困難をきたしているかどうかを第三者が知ることは容易ではない．なぜならば，プログラム理解は脳内における種々の認知プロセス (注意，記憶，推論など) のはたらきで実現される内面的な活動だからである．

これまで，開発者の内面的活動であるプログラム理解の進捗や成功・失敗を計測するため，インタビューやテスト，あるいは思考内容の発話を伴うアプローチが多く用いられてきた [9] [5]．ただし，これらの手法では開発者の状態を即時に知ることが出来ないか，あるいは即時性を向上するために頻繁な作業への割り込みを必要とし，プログラム理解そのものに支障を来すという問題がある [10]．

一方，神経科学や認知科学の分野においては，脳波や脳血流といった脳周辺情報を用い，非侵襲的に知的活動を計測するアプローチが広く用いられている [2]．これらの研究では，知的活動の種類や量と，脳の特定部位の活動を結び付ける試みがなされている．近年，プログラム理解の分野においても，こうした神経科学

Measuring Difficulty of Program Comprehension Based on Brain Activation
Takao Nakagawa, Kenichi Matsumoto, 奈良先端科学技術大学院大学情報科学研究科, Graduate School of Information Science, Nara Institute of Science and Technology.
Yasutaka Kamei, Naoyasu Ubayashi, 九州大学大学院システム情報科学研究院, Graduate School and Faculty of Information Science and Electrical Engineering, Kyushu University.
Hidetake Uwano, 奈良工業高等専門学校情報工学科, Department of Information Science, National Institute of Technology, Nara College.
Akito Monden, 岡山大学大学院自然科学研究科, Graduate School of Natural Science and Technology, Okayama University.
コンピュータソフトウェア, Vol.29, No.1 (2012), pp.78–84.

分野の知見を応用し，プログラム理解時の脳活動を分析する試みが行われつつある [17] [18]．これらの手法を利用し，作業を阻害することなくプログラム理解中の困難さを計測できれば，プログラム理解に行き詰った学習者にアドバイスを与えるといったサポートが行えるため，特に初学者に対するプログラミング教育に寄与できると期待される．

本論文では，プログラム理解の脳計測による定量化の第一歩として，人間がプログラム理解に困難を感じている状態を NIRS (Near InfraRed Spectroscopy) によって判別することを目的とし，実験によって以下のリサーチクエスチョン (RQ) を確認する．

> RQ1：プログラム理解に困難が生じている状態を，理解時の脳血流によって判別できるか？

開発者にとってプログラムの理解難易度が高い場合には，理解に必要な知的活動の量が増大し，記憶の操作や意識の統合に代表される高次の認知機能を担うとされる脳前部 (前頭前野) [24] が活発化すると考えられる．このことから，異なる難易度のプログラムを読んだ際の被験者の前頭前野の脳血流を計測することにより，RQ1 の検証を試みる．

> RQ2：プログラム理解の困難さの度合いを，脳血流計測によって区別できるか？

RQ2 では，難易度の異なる (3 段階) プログラム理解の区別を試みる．

> RQ3：脳血流計測の結果は，被験者が感じる主観的な難易度を反映しているか？

RQ3 では，被験者アンケートによる主観的な難易度と脳血流との関係を明らかにする．

本論文では妥当性・有用性の向上を目的として，我々の先行研究 †1(レター論文) に対して実験設計の改良，および追加実験 (RQ2, RQ3) を行った．

本論文の主な貢献は次の通りである．

- **新たな実験による結果の信頼性向上**

  **先行研究の課題**　先行研究では，被験者の人数が 10 名と少なく，統計的な議論を行えなかった．実験結果にも仮説と逆の傾向を示す被験者が 2 名見られ，信頼性に疑問が残っていた．

  **課題への対応**　本研究では，実験結果の信頼性を向上するため，以前と異なる被験者を 20 名採用した新たな実験を実施し，脳血流計測の結果と課題難易度の間に有意な関係が見られることを，検定により確認した．

- **ノイズ除去による結果の信頼性向上**

  **先行研究の課題**　先行研究では，頭部動作や長期的な血流のトレンドによるノイズを除去できなかった．このため,「タスクと本質的に関係のない変化量が結果に含まれる」や,「タスクにあわせて被験者が頭部の角度を変えることで測定値に影響する」などの可能性があった．

  **課題への対応**　頭部動作に対して CBSI 法によるノイズ除去を，長期的な血流変化に対して EMA(指数移動平均) によるノイズ除去を行うことで，実験結果の信頼性・妥当性を向上させた．

- **異なる難易度における脳血流の分析**

  **先行研究の課題**　先行研究では，理解対象プログラムの難易度が 2 段階のみであり,「やや難しいプログラム」の理解と「非常に難しいプログラム」の理解の脳活動を区別できるか否かは不明であった．

  **課題への対応**　RQ2 として，実験タスクの難易度を 3 段階 (easy, medium, difficult) 設けるとともに，脳活動量との関係を分析した．

- **主観的難易度と脳血流との関係の分析**

  **先行研究の課題**　実験者が設定したプログラムの難易度と，被験者が感じている理解の難易度は必ずしも一致しない可能性があった．

  **課題への対応**　RQ3 として，被験者アンケートによる主観的難易度と脳血流との関係を分析した．

†1 中川尊雄, 亀井靖高, 上野秀剛, 門田暁人, 松本健一, “プログラム理解の困難さの脳血流による計測の試み,” コンピュータソフトウェア, 31(3), pp. 270-276, 2014. (レター論文)

## 2 関連研究と背景技術

### 2.1 プログラム理解の計測

プログラム理解を対象とした従来研究では，被験者の主観や精神状態，作業の進捗状況などを測定対象とした心理学的なアプローチが用いられてきた．Parnin らは複数のプロジェクトに携わる開発者のプログラムに対する記憶について考察し，記憶の精度・内容が時間経過でどのように変化するか実験とインタビューから分析した [14]．中村らは変数の記憶と想起に注目したプログラム理解のモデルによって，理解にかかる時間を近似できることを示した [13]．また，石黒らは中村らのモデルを改良し，繰り返し参照される記憶を想起しやすくなる効果 (リハーサル効果) が，理解にかかる時間に影響を与えた可能性について述べている [25]．

プログラム理解の過程を被験者の作業を阻害することなくリアルタイムに計測するための手法として，皮膚抵抗値を用いた精神的な緊張の計測 [26] や視線移動の計測 [16] など生体情報を用いた手法が提案されている．Fritz らはプログラム理解時の視線，脳波，筋電を計測し，機械学習によってタスクの難易度を推定する実験を行い，多数のセンサをどのように組み合わせると高い精度で予測が可能か調べた [7]．

近年では脳の周辺情報 (脳血流，脳波) を測定することでプログラム理解・作成に含まれる認知プロセスや開発者のストレスを推定する研究が行われている．

Siegmund らはプログラム理解中に活性化する脳領域を fMRI (functional Magnetic Resonance Imaging) で計測し，問題解決，記憶，文章理解を関連する認知プロセスとして挙げた [17]．

幾谷らは，NIRS を用いて，変数と条件分岐の差が脳活動に与える影響を調査し，変数操作が含まれるコードの読解時に前頭極の脳活動が活発化することを報告している [8]．

プログラム理解時における脳の活性化度合いを計測する点でこれらの研究は本稿と密接に関連しているが，以下の点で異なっている．Siegmund らは，姿勢や体動への制限が大きい fMRI を計測に用いており，実際の開発現場への適用を対象としていない．加えて，難易度の異なる課題の比較を対象としていない．本研究では，体動や姿勢に制限が少ない NIRS を用い，実環境に近い状態で，難易度の異なるプログラムを理解する際に生じる困難を判別することを目的にしている．幾谷らは数値演算，変数操作，条件分岐の三種類の構文を独立に含む 10 行未満の短いコード片を対象としているが，本研究では 17〜32 行の，一定の目的を持った関数を対象としている．また，プログラムの難読化を用い，被験者の主観評価を含めて難易度の違いの判別を目的にしている点で，幾谷らの研究とは立場が異なる．

### 2.2 脳活動計測による認知プロセスの分析

神経科学分野においては，各種の知的活動と脳の関係を調査する際，脳活動に関係する生体情報 (脳周辺情報) を計測することが一般的である．脳周辺情報の計測に用いられる一般的な手法・機器として，脳の血流動態を測定する fMRI および NIRS，脳表面の電位 (脳波) を測定する EEG (Electroencephalogram) などがある．

本研究で用いる NIRS は，脳が活動した際の神経活動に応じて酸素を供給するために脳血流量とその酸化度合が上昇する現象を利用して脳活動量や部位を特定する手法である．血中の酸化ヘモグロビン (oxy-Hb) と脱酸化ヘモグロビン (deoxy-Hb) の吸光特性が異なることを利用し，頭皮表面に照射した近赤外光の反射成分を検知することで脳表層における活動を計測する．fMRI や MEG (Magnetoencephalography) などの他装置と比較して，計測に必要な準備が少なく取り付けが容易で，姿勢や体動の制限が少ないほか，時間分解能 (計測値の時間方向への精度) が高いという利点がある．一方で，計測値が頭蓋や皮膚の厚さに影響を受けるため，そのままでは装置を着脱した前後の比較や被験者間の比較ができない．

脳活動計測を用いた多くの研究では特定の知的活動 (例えば暗算 [21] や自然文読解 [11]) や身体動作と，特定の脳部位 (前頭前野，頭頂葉，側頭葉) における活動量の関連を調査することに焦点を当てている．例えば，Zago らは暗算に関連する脳部位について，異なる難易度でどのような差分があるかを PET で計測

している [21]．Cabeza らは fMRI や PET を用いた 275 件の研究をサーベイし，認知プロセスと脳部位の対応を詳細に報告している [2]．こうした基礎研究では，特定の単語の発話や 1 桁の掛け算といった単純な課題と関係する脳部位を詳細に特定することが目的となっており，実験には空間分解能 (計測値の空間的な精度; 脳領域を特定する能力) が高く時間分解能が低い fMRI や PET がよく用いられる．

本研究では，プログラム理解と関係する認知プロセスである記憶の操作や推論の際に活性化する前頭前野 [20][24] に注目し，NIRS を用いた計測を試みる．NIRS は計測時の姿勢に制限が少なく，端末の前に着座した状態での計測が可能であるため，他の計測装置と比べてプログラム理解の計測に適している．また，fMRI や PET に比べて軽量かつ安価であることから，現場への適用や長時間の計測にも適している．

## 3 実験

異なる難易度のプログラム理解を試みる際の脳活動を NIRS で計測する被験者実験を行った．被験者は奈良先端科学技術大学院大学，奈良工業高等専門学校，九州大学の学部生・大学院生計 20 名で，全員が 20 歳から 24 歳の男性で，プログラミング経験が 3 年以上であった．

### 3.1 手順

実験において被験者は，NIRS 装置を装着した状態で，理解難易度の異なる 2 種類の暗算タスクと 4 種類のプログラム理解 (練習，easy，medium，hard) タスクを 1 つずつ，計 6 タスクを実施する．

暗算タスクには，難易度ごとに前頭前野の脳活動に差があると報じた文献 [21] で用いられたものを利用する．本研究では，事前に知られた 2 種類の暗算タスクの間に見られる脳活動の差と，難易度の異なるプログラム理解タスクの間に見られる脳活動の差を比較し，その一致度を見ることで，後者の結果の信頼性を補強することをはかる．

プログラム理解タスクには，機能の異なるプログラムを用いる．各プログラムに対して理解しやすさを変更した 3 つのソースコード (easy, medium, difficult) を作成し，タスクとして用いる．学習・順序効果の影響を防ぐため，easy，medium，difficult で同じ機能のプログラムが提示されないよう配慮し，実施する難易度・機能の順番も均等にランダム化する．

以下に実験手順を示す．実験の最初と各タスクの間には，前後のタスクが計測値に与える影響を抑えるために，紙に印刷した十字の模様を二分間注視してもらう．

- 簡単な暗算 (2 分)
- 難しい暗算 (2 分)
- 練習問題 (10 分もしくは終了次第)
- プログラム理解 1(10 分もしくは終了次第)
- プログラム理解 2(10 分もしくは終了次第)
- プログラム理解 3(10 分もしくは終了次第)
- 計測終了，アンケートの実施

### 3.2 タスク

**プログラム理解タスク:** プログラム理解タスクにおいては，理解中に生じる困難によって脳活動に変化があるかを調査するため，被験者がプログラム理解に困難を感じる状態と感じない状態を人為的に誘発する．理解の困難さをつくりだす要因や，理解中に取られる戦略には様々なものがあり，本タスクでは困難を誘発するよう，これらに制約を加える．

理解が困難となる要因には，アルゴリズム自体の複雑さや，保守の過程におけるコードの劣化などが考えられる．本タスクでは，小規模プログラムにおけるアルゴリズムの複雑さに注目し，その再現のために，難読化手法によって制御フローの複雑性を変化させる．一般に，制御フローが複雑であるほどプログラム理解が困難となり，バグ混入の可能性が高まることが知られている [1]．

また，プログラム理解のために開発者がとる行動（理解戦略）には，モジュール間の呼び出し関係を調べる方法 [6] や，データの流れを追跡する方法 [6]，プログラムの Goal と Plan に対する仮定を立てて検証する方法 [23]，プログラムの実行過程を追跡する方法（メンタルシミュレーション）[15] などがある．これらの手法は，何れか一つでプログラム理解を完全に達成できるものではないが，状況に応じて様々な戦略が

```
A[0][0][0] = 97, A[0][0][1] = 48, A[0][1][0] =52
A[0][1][1] = 71, A[1][0][0] = 17, A[1][0][1] =64
A[1][1][0] = 11, A[1][1][1] = 32, A[2][0][0] =20
A[2][0][1] = 22, A[2][1][0] = 48, A[2][1][1] =86

N = 3, M = 2, L = 2
```

```
int func(int ***A, int N, int M, int L){
    int i,j,k;
    int p;
    p=A[0][0][0]; (1)
    for(i = 0; i < N; i++){
        for(j = 0; i < M; j++){
            for(i = 0; i < N; i++){
                if(A[i][j][k] < p) p = A[i][j][k]; (2)
            }
        }
    }
    return p; (3)
}
```

**図 1　実験で用いたソースコードと提示した引数の例**

**表 1　解答用紙の記入例**

| N | M | L | i | j | k | p | 位置 |
|---|---|---|---|---|---|---|---|
| 3 | 2 | 2 | | | | 97 | (1) |
| | | | 0 | 0 | 1 | 47 | (2) |
| | | | 1 | 1 | 0 | 11 | (2) |
| | | | 1 | 0 | 0 | 17 | (2) |
| | | | | 1 | 0 | 11 | (2) |
| | | | 2 | 1 | 1 | | (3) |
| | | | 3 | 2 | 2 | | (3) |

採用される．本タスクでは，制御フローの理解に多く用いられ，プログラムがどのように動くかを知る上で必須となる [4] メンタルシミュレーションを採用する．

上記の条件を踏まえ，本タスクでは紙に印刷された 17〜32 行の C 言語の関数と引数のペアを元に，被験者が動作をメンタルシミュレーションし，その過程を解答用紙に記入する．図 1 にタスクで用いるプログラムの例を，表 1 に解答用紙の記入例を示す．

被験者はメンタルシミュレーションが目印 (ソースコードの右側に書かれた番号) の行に到達するたびに，各変数の値と目印の番号を解答用紙に記入する．実験者は随時回答を確認し，値がすべて正しければシミュレーションを続けてもらう．そうでなければ，前回の正答時点からシミュレーションをやり直し，再度解答用紙に記入してもらう．目印はループ中に含まれることもあるため，同じ目印に対して複数回の回答をすることもある．タスクは，メンタルシミュレーションが終了するか，10 分の制限時間を迎えた時点で終了とする．

タスクで用いるプログラムの概要を表 2 に示す．easy で提示されるソースコードはそれぞれの機能の一般的な実装であり，medium，difficult は，easy に制御フローを複雑にする難読化手法 [12] を適用することで作成する．medium に用いる手法は，プログラム内の文を複製し，条件分岐のリンクを付け替えることで制御構造を複雑化する．difficult に用いる手法ではこれに加え，ループ条件式のうち，通常は定数である部分を変化させることで，文の評価順を極めて複雑にする．

**暗算タスク:** 暗算タスクは，一桁どうしの掛け算 (簡単な暗算) と，二桁どうしの掛け算 (難しい暗算) からなる．Zago らの研究 [21] によると，簡単な暗算は「九九」のように計算結果を長期記憶から想起するだけで，実際の計算は行われない．一方で，難しい暗算においては，長期記憶の想起に加え，実際の計算を行うために一時的な記憶領域が利用され，簡単な暗算に比べて前頭前野が活発化する．

### 3.3　アンケート

各タスクの終了後，被験者に対して質問用紙を用いたアンケートを実施する．アンケートでは，提示された課題それぞれの難しさを「簡単」から「難しい」まで 5 段階のリッカート尺度で解答してもらうほか，日頃プログラムを読む頻度や，プログラミング経験，利き腕，九九を覚えているか尋ねた．また，アンケートには自由記述欄を用意する．

なお，アンケートの記入時，被験者が課題の内容を

**表 2　使用したプログラムの概要**

| 名前 | 難易度 | 機能 |
|---|---|---|
| eA | easy | 最小値の探索 |
| eB | easy | 数値の合算 |
| eC | easy | 特定文字の数え上げ |
| mA | medium | 最小値の探索 |
| mB | medium | 数値の合算 |
| mC | medium | 特定文字の数え上げ |
| dA | difficult | 最小値の探索 |
| dB | difficult | 数値の合算 |
| dC | difficult | 特定文字の数え上げ |

忘れた場合を考慮し，タスクで用いた問題用紙を実施順に並べて被験者に提示する．

### 3.4 実験環境

実験には日立製ウェアラブル光トポグラフィWOT-220 を用いる．本装置は前頭葉の脳血流量を測ることが可能で，また，他の NIRS 装置と比較して軽く，装着が容易であるという特徴を持っている．図 2 に装置と装着時の外観を示す．

実験は被験者 1 名と実験者 1 名のみが居る静かな部屋において，椅子に座った状態で行われた．姿勢の変化によるノイズを防ぐため，ソースコードを印刷した紙と回答用紙を書見台に置き，楽な姿勢で読めるよう椅子の位置や高さを調整してもらう．また，頭部の動きによる血流変化を防ぐため，事前に頭を激しく動かさないよう被験者に伝える．

### 3.5 分析

本研究では，NIRS で計測した値に対してノイズ除去と標準化を適用して求められる標準化 oxy-Hb を用いて，各被験者ごとのタスク種類ごとにおける脳活動量の差を議論する．

#### 3.5.1 ノイズ除去

**装置に由来するノイズの除去:**装置由来のノイズとして，a) 測定値に恒常的に含まれるバイアスと，b) 工学的特性によるスパイクノイズがあげられる．
このうち前者は，測定前のキャリブレーションによって取り除く．後者のスパイクノイズは，標準移動平均 (SMA; Simple Moving Average) をとることで除去する．本実験における最も短いタスク (2 分) に対して十分短い 4 秒の SMA を計算する．機器の測定周波数は 5[Hz] なので，oxy-Hb と deoxy-Hb の計測値に対し，式 (1) と式 (2) を用いて n=20 での信号平滑化を行う．

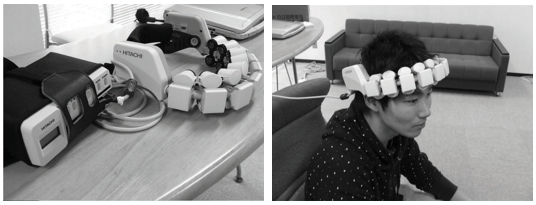

**図 2 装置と装着時の外観**

$$oxy_{SMA}(t) = \frac{1}{n}\sum_{k=1}^{n} oxyHb(t) \tag{1}$$

$$deoxy_{SMA}(t) = \frac{1}{n}\sum_{k=1}^{n} deoxyHb(t) \tag{2}$$

**生体に由来するノイズの除去**:生体由来のノイズとして，心拍，呼吸，血圧や体調に依存する変化があげられる．このうち心拍や呼吸などによるノイズは短周期な一定の変動であるため SMA によって除去される．一方，血圧や体調の変化などを含むタスク実施時間より長周期なノイズについては，原信号から指数移動平均 (EMA; Exponential Moving Average) を減算する手法 [19] を用いる．本実験における最も長いタスク (10 分) の 2 倍にあたる 20 分の EMA を計算する．機器の測定周波数は 5[Hz] なので，$oxy_{SMA}$ と $deoxy_{SMA}$ に対し，式 (3) と式 (4) を用いて n=6000 での信号平滑化を行う．

$$\begin{aligned} oxy_{MA}(t) = {} & \frac{1}{n} oxy_{SMA}(t) \\ & + (1 - \frac{1}{n}) oxy_{MA}(t-1) \end{aligned} \tag{3}$$

$$\begin{aligned} deoxy_{MA}(t) = {} & \frac{1}{n} deoxy_{SMA} \\ & + (1 - \frac{1}{n}) deoxy_{MA}(t-1) \end{aligned} \tag{4}$$

**長期計測に由来するノイズの除去**:NIRS の測定値は頭部や体の動きでセンサー位置が変化することでノイズが加算される．本実験では，問題用紙の読解や解答用紙への記入に合わせて頭部が上下する可能性がある．そこで，Cui らによる CBSI 法 [3] を採用し，頭部・身体動作に起因するノイズの除去を試みる．
CBSI 法は，脳活動があった際，oxy-Hb と deoxy-Hb の値が反比例の関係を示すのに対し，身体動作に端を発する計測値変化においては oxy-Hb と deoxy-Hb が比例関係を示すことを利用したノイズ低減手法で

ある．

分析に際しては，SMA と EMA による信号平滑化を行った oxy-Hb と deoxy-Hb(式中 $oxy_{MA}$，$deoxy_{MA}$) に対して，次の式 (5) を適用し，真の oxy-Hb と deoxy-Hb(式中 $oxy_{CBSI}$，$deoxy_{CBSI}$) を求める．

$$\begin{aligned} \alpha &= \frac{sd(oxy_{MA}(t))}{sd(deoxy_{MA}(t))} \\ oxy_{CBSI}(t) &= \frac{1}{2}(oxy_{MA}(t) - \alpha deoxy_{MA}(t)) \\ deoxy_{CBSI}(t) &= -\frac{1}{\alpha} oxy_{CBSI}(t) \end{aligned} \tag{5}$$

#### 3.5.2 計測値の標準化

NIRS による計測値は，計測開始時からの脳活動の相対的な変化量を表すため被験者間の比較が難しい．そこで，本稿ではある時点 t における脳活動の強さを，$oxy_{CBSI}$ 値の平均と分散を用いて正規化した activation(次式) の形式で表現することで被験者間の比較を行う．

$$activation(t) = \frac{oxy_{CBSI}(t) - mean(oxy_{CBSI})}{sd(oxy_{CBSI})} \tag{6}$$

$activation(t)$ は，時刻 t における脳活動の強さを表す．ここで用いる $oxy_{CBSI}(t)$ の平均値と標準偏差は，被験者ごとに計算されるため，activation は被験者ごとに平均 0, 分散 1 の形に正規化されたものとなる．

## 4 結果

20 名の被験者のうち，3 名の被験者は実験中に装置が大幅にずれたため結果から除外した．

### 4.1 各タスクごとの脳活動値

本実験は，1 タスクの時間が最長で 10 分と長く，脳が活性化するタイミングが被験者ごと，タスクごとに異なることから，平均値や波形による議論が難しい．そのため，RQ1 の検証にあたって，実施したタスク難易度ごとの activation の分布を表した箱ひげ図を図 3 に示し，議論を行う．

図からは，17 人中 16 人について easy より difficult の中央値が高いことが読み取れる．また，17 人中 15 人について easy より medium の中央値が高いことも読み取れる．17 人中 16 人，ならびに 17 人中 15 人という結果について，正確二項検定を実施したところ，いずれも $p < 0.01$ で有意な偏りであった．このことは，「プログラム理解に困難が生じている状態を，理解時の脳血流計測によって計測できる」という仮説を支持する．

> RQ1 への回答：理解難易度の異なる様々なプログラムを読んだ際，被験者に生じる困難を，脳血流計測によって判別できる．

ただし，本研究で採用した難読化手法 [12] で生成されたプログラムの難易度が，現実世界におけるプログラムの難易度を反映しているかどうかは不明であり，今後の検討課題である．

また，同様に平均値を見た場合も，17 人中 16 人について easy より difficult が，17 人全員について easy より medium が高かった．一方，medium と difficult の平均値を比較すると，difficult の方が高かったものが 10 名，medium の方が高かったものが 7 名であった．

### 4.2 被験者に実施したアンケートの結果

被験者に実施したアンケートの結果得られた，被験者が各課題に対して感じた主観的な難易度 (5 段階, リッカート尺度) を表 3 に示す．表からは，被験者が高難易度タスクほど難しく感じたと回答していることが読み取れる．

難易度別の三群間に有意な差があるかどうかをクラスカル＝ウォリス検定によって確かめた結果，群間の差は $p < 0.01$ で有意であった．

### 4.3 暗算タスクの脳活動値

NIRS 機器の計測値や，その加工 (activation の算出) についての妥当性を調査すべく，関連研究 [21] において，難易度が高いと脳活動が活発化することが報告されている暗算課題について，その差を確認できるかどうかを調べた．

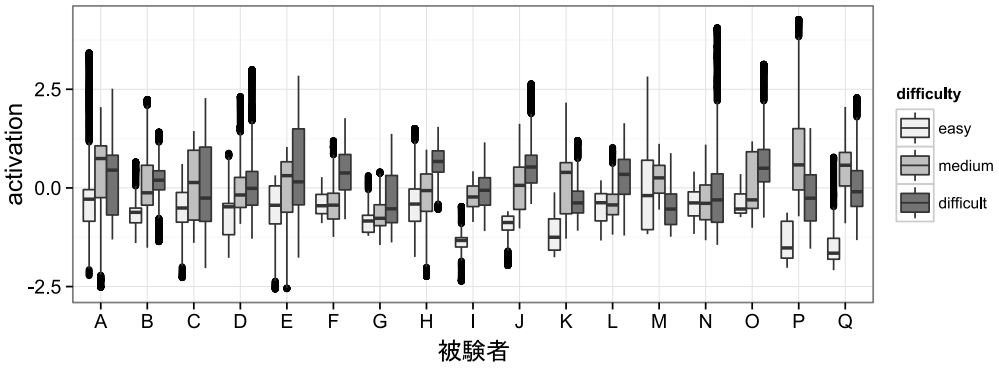


**図 3 被験者ごとの難易度別 activation**

**表 3 各課題に対する主観的難易度**

| 被験者 | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | mean |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| easy | 3 | 1 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 1 | 1 | 1 | 1 | 2 | 2 | 1 | 1.39 |
| medium | 2 | 2 | 3 | 2 | 4 | 2 | 4 | 2 | 4 | 2 | 2 | 5 | 3 | 2 | 5 | 4 | 3 | 3.00 |
| difficult | 5 | 5 | 5 | 4 | 3 | 5 | 3 | 4 | 5 | 5 | 4 | 4 | 4 | 4 | 4 | 4 | 5 | 4.22 |

その結果，17 人中 14 人について，簡単な暗算課題実施中より，難しい暗算課題実施中の activation が高いことが確認された．17 人中 14 人という結果ついて，正確二項検定を実施したところ，$p < 0.05$ で有意であった．

本結果は，17 名中 3 名の例外を含むものの，NIRS 機器による activation の計測が一定の妥当性を持つことを支持するものと考えられる．

## 5 考察

### 5.1 異なる難易度における脳血流の分析

設定したタスク難易度と脳活動量の関係を分析するために，アンケートと同じく難易度別の三群間に有意な差がみられるかクラスカル＝ウォリス検定によって確かめた結果，群間の差は $p < 0.01$ で有意であった．このことは，脳血流の分布から難易度の差を判別できることを示す．その一方で，medium と difficult の二群間の中央値の差について，ウィルコクソンの符号付き順位和検定を実施したところ，有意な差は見られなかった．

3.2 節で述べたように medium の条件は「制御構造に対する難読化が施されている」という点で difficult と重なっているため，medium と difficult の間に差が出にくかったと考えられる．すなわち，制御フローに対する難読化が，プログラムの処理内容に対するトップダウンな理解を防ぐことで，結果的に複雑度に関わらず被験者は「記述内容を一行ずつ解釈する」ことになり，単位時間あたりに要する思考の質に大きな差があらわれなかった可能性がある．

この結果は，プログラム理解中の開発者に対する脳血流計測では，ある一定以上負荷が高くなると，それ以上の負荷は判別できないことを示す．

> RQ2 への回答：プログラムの構造を簡単に理解できたかどうかを判別できる．ただし，トップダウンな理解が難しい課題における，極度の負荷とそれ以下の負荷の判別はできない．

### 5.2 被験者の主観的難易度と脳活動量の関係

ここまでは，著者が実験設定時に決定した課題の難易度 (easy, medium, difficult) と脳活動量の関係を論じてきたが，これらの課題難易度は必ずしも被験者が感じた主観的な難易度と一致しない．そこで，被験者が感じた主観的難易度と，脳活動量の間にどのような関係が見られるか調べた．

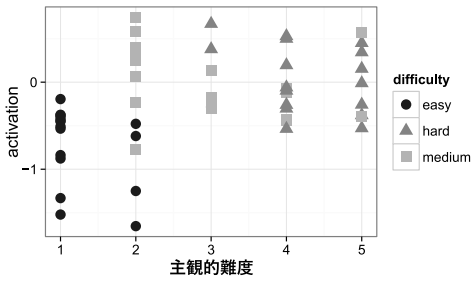


**図 4　主観的難易度と activation の中央値**

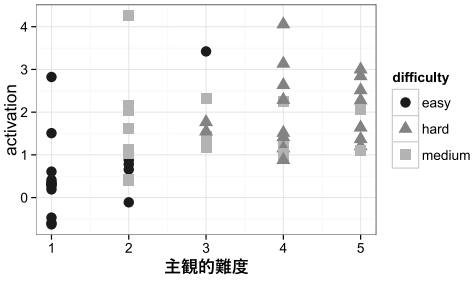


**図 5　主観的難易度と activation の最大値**

図 4 は被験者の感じた主観的難易度 (5 段階) を横軸，activation の中央値を縦軸にとった散布図である．全体として課題の難易度が高くなるほど主観的評価と activation の値が高くなる比例関係が見られる．activation と主観的難易度の関係について，Spearman の相関係数をとったところ，相関係数 0.42($p < 0.01$) で有意な相関がみられた．

easy に着目すると，主観的難易度と activation が一致して共に低く，他の難易度と分かれていることがわかる．一方で，主観的難易度が 2 と回答された課題に着目すると，easy の activation が低く，medium が高い．これは，主観的難易度が同様の場合であっても activation を計測することで難易度の異なる課題を判別できる可能性を示唆する．

また，最も activation が活発だった時の値，すなわち各タスク実施中の activation の最大値と主観的難易度の関係を示した散布図を図 5 に示す．最大値を見ると，medium より difficult のほうが activation の値が高い場合が多い．Spearman の相関係数は 0.59($p < 0.01$) と中央値を用いた場合よりも強い，有意な相関がみられた．

このことは，もっとも難易度の高い部分を読解する必要がある時点においては，difficult のほうが medium より高い activation を示す可能性があることを示唆する．ただし，最大値については，easy でも高い値が見られ，全体的にばらつきが大きいため，activation の最大値が課題遂行中の困難を示すか確かめるには至らなかった．

RQ3 への回答：activation の中央値は主観的難易度と有意な相関があり，特に 5 段階評価で 2 を下回る場合，明瞭に区別できる．最大値を見ると，さらに相関が強いが，ばらつきが大きい．

## 6　妥当性への脅威

本章では，本研究の結論の妥当性への脅威について述べる．

### 6.1　内的妥当性への脅威

まず，内的妥当性への脅威として，被験者実験において，課題難易度が低いほど課題実施時間が短く終り，課題難易度が高いほど制限時間に到達していた傾向が見られた．このことが結果における activation の難易度間の差に影響を与えている可能性がある．ただし，medium と difficult を比較すると，課題実施時間に差が見られる一方，activation に差は見られず，実施時間が activation の値，ならびに実験結果に与える影響は小さいものと考えられる．

また，被験者はそれぞれプログラミング経験や利用言語が異なり，コードの読解について個人間で慣れの影響があることが考えられる．本研究では，このような慣れの影響を排除するため，練習課題の実施中にソースコードの構文や読み方についての質問を受け付けたほか，主観評価においてはいずれの被験者も difficult に 3 以上の値をつけていたことから，難易度設定についての妥当性は確保されているものと考えている．

### 6.2 外的妥当性への脅威

本論文では，被験者がプログラム理解に困難を感じる状態を誘発するため，実験タスクではアルゴリズムの理解しにくさに注目し，制御フローの難読化を用いて難易度を調整している．また，理解戦略も，制御フローやデータフローの理解に用いられるメンタルシミュレーションに限定している．一方で，理解を困難にする要因には，アルゴリズムの複雑性以外にもモジュールの凝集度・結合度，不適切な変数名やコメントなど様々な種類があり，理解戦略もそれに応じて多様なものが用いられる．

これらの制約のため，現時点における本研究の成果は，小規模プログラムにおけるアルゴリズムの理解が困難な状況の検出・支援に限定され，また，現実のプログラムの読みにくさを反映しているとは限らない．より大規模で現実的なプログラムの理解を想定し，編集に伴うモジュールの役割変化，不適切なコメントに代表されるコード劣化など，多様な困難を誘発する状況下での実験を行うことが，今後の発展的課題となる．

また，本研究では実験の被験者がすべて学生であった．このことは，プログラム開発に熟練した開発者を計測する場合などを想定した場合，外的妥当性への脅威となりうる．今後は，より実際の開発環境に近い実験設定，例えば開発支援ツールやコードハイライトなどが有効な環境で，一定以上のサイズのプログラムを書いた経験者に限定した実験を実施することにより，本手法の有効性を検証したい．

## 7 おわりに

プログラム理解はソフトウェア開発のさまざまな場面で必要となる活動であり，理解がうまくいかない状態でコードレビューなどが行われた場合，多量の欠陥がソフトウェア中に混入する原因となりうる．プログラム理解がどの程度スムースに進んでいるかを，作業への割り込みなしに把握できれば，レビュー中のアドバイスや人員交代などにより，効率的なレビュー・開発の実現が可能となる．

本論文では，開発者のプログラム理解に困難が生じている状態の定量的な判別を目指し，脳血流計測の利用を提案した先行研究における実験に改良を加え，信頼性や有効性の拡張を図った．

先行研究においては，10 名中 8 名の被験者において難読化したプログラムの理解中に脳活動が活発になる傾向が明らかとなったが，2 名については逆の傾向が観測された．また，二種の難易度のみを対象に実験を実施したため，脳血流計測の測定結果がどの程度の粒度でプログラム理解中に生じる困難さを測定できるのかが不明であった．

本研究では，先行研究における提案の信頼性・有効性を検証し，強化するため，先行研究よりも堅牢な実験計画を採用した．実験には，被験者を休息させる Rest タスクの採用，三段階目の難易度である medium 課題の採用，関連研究において NIRS による計測実績がある暗算課題の追加，そして被験者の追加といった改良を加えた．

実験の結果として 17 名の被験者のうち 16 名について，difficult 課題遂行中の脳活動が easy 課題遂行中に比べて活発であることがわかった．このことは，「プログラム理解に困難が生じている状態を，理解時の脳血流計測によって計測できる.」という仮説を支持するものである．ただし，本研究ではアルゴリズムの複雑さに起因する困難さを想定し，機械的な難読化手法を適用したプログラムを読んでもらう実験設定を採用した．そのため，本研究の結果がその他の要因に起因する困難さや，現実世界でのプログラムの複雑性を必ずしも反映できているかは確認できておらず，今後の重要な課題となる．

また，difficult 課題遂行中の脳活動量と medium 課題遂行中の脳活動量には有意な差が見られなかった．そのため，プログラム理解中に直面した困難について，その困難さの度合いまでは本手法では区別できない可能性があることが明らかとなった．

本研究で示した結果は，いずれも実験設定に示した環境のもとで得られたものであり，その適用可能範囲は数あるプログラム理解活動のうち，一部に限られる．そこで，今後の展望として，支援を必要とするプログラム理解活動という観点から問題を整理したうえで，計測手法の適用先や有効性を検討していくことが課題となる．例えば，プログラム実行，デバッ

ガ等のツールの利用，設計書の参照等を含むプログラム理解活動に対して，脳活動計測による具体的な支援を検討していきたい．

具体的な方針としては，プログラムの実行やデバッガに代表されるツールの利用，あるいは設計書の参照など，他の理解活動について考慮した調査が求められる．

## 参 考 文 献


[1] Arisholm, E. and Briand, L. C.: Predicting fault-prone components in a java legacy system, *Proceedings of the 2006 ACM/IEEE international symposium on Empirical software engineering*, ACM, 2006, pp. 8–17.

[2] Cabeza, R. and Nyberg, L.: Imaging cognition II: An empirical review of 275 PET and fMRI studies, *Journal of cognitive neuroscience*, Vol. 12, No. 1(2000), pp. 1–47.

[3] Cui, X., Bray, S., and Reiss, A. L.: Functional near infrared spectroscopy (NIRS) signal improvement based on negative correlation between oxygenated and deoxygenated hemoglobin dynamics, *NeuroImage*, Vol. 49, No. 4(2010), pp. 3039–3046.

[4] Détienne, F.: Expert programming knowledge: a schema-based approach, *Psychology of programming*, (1990), pp. 205–222.

[5] Ericsson, K. A. and Simon, H. A.: Verbal reports as data., *Psychological review*, Vol. 87, No. 3(1980).

[6] Fix, V., Wiedenbeck, S., and Scholtz, J.: Mental Representations of Programs by Novices and Experts, *Proceedings of the INTERCHI '93 Conference on Human Factors in Computing Systems*, Amsterdam, The Netherlands, The Netherlands, IOS Press, 1993, pp. 74–79.

[7] Fritz, T., Begel, A., Müller, S. C., Yigit-Elliott, S., and Züger, M.: Using psycho-physiological measures to assess task difficulty in software development, *Proceedings of the 36th International Conference on Software Engineering (ICSE 2014)*, ACM, 2014, pp. 402–413.

[8] Ikutani, Y. and Uwano, H.: Brain Activity Measurement during Program Comprehension with Nirs, *International Journal of Networked and Distributed Computing*, Vol. 2, No. 4(2014), pp. 259–268.

[9] Karahasanović, A., Hinkel, U. N., Sjøberg, D. I., and Thomas, R.: Comparing of feedback-collection and think-aloud methods in program comprehension studies, *Behaviour & Information Technology*, Vol. 28, No. 2(2009), pp. 139–164.

[10] Karahasanović, A., Levine, A. K., and Thomas, R.: Comprehension strategies and difficulties in maintaining object-oriented systems: An explorative study, *Journal of Systems and Software*, Vol. 80, No. 9(2007), pp. 1541–1559.

[11] Keller, T. A., Carpenter, P. A., and Just, M. A.: The Neural Bases of Sentence Comprehension: a fMRI Examination of Syntactic and Lexical Processing, *Cerebral Cortex*, Vol. 11, No. 3(2001), pp. 223–237.

[12] Monden, A., Takada, Y., and Torii, K.: Method for Scrambling Programs Containing Loops, *IEICE Trans. on Information and Systems*, Vol. 80, No. 7(1997), pp. 644–652.

[13] Nakamura, M., Monden, A., Itoh, T., Matsumoto, K., Kanzaki, Y., and Satoh, H.: Queue-based cost evaluation of mental simulation process in program comprehension, *Proceedings of 9th IEEE International Software Metrics Symposium (METRICS 2003)*, Sep. 2003, pp. 351–360.

[14] Parnin, C.: A cognitive neuroscience perspective on memory for programming tasks, *Proceedings of 22nd Annual Meeting of the Psychology of Programming Interest Group (PPIG)*, 2010.

[15] Pennington, N. and Grabowski, B.: The tasks of programming, *Psychology of programming*, Vol. 307(1990), pp. 45–62.

[16] Sharif, B., Falcone, M., and Maletic, J. I.: An eye-tracking study on the role of scan time in finding source code defects, *Proceedings of the Symposium on Eye Tracking Research and Applications 2012 (ETRA 2012)*, 2012, pp. 381–384.

[17] Siegmund, J., Brechmann, A., Apel, S., Kästner, C., Liebig, J., Leich, T., and Saake, G.: Toward measuring program comprehension with functional magnetic resonance imaging, *Proceedings of the ACM SIGSOFT 20th International Symposium on the Foundations of Software Engineering (FSE '12)*, ACM, 2012, pp. 24:1–24:4.

[18] Siegmund, J., Kästner, C., Apel, S., Parnin, C., Bethmann, A., Leich, T., Saake, G., and Brechmann, A.: Understanding Understanding Source Code with Functional Magnetic Resonance Imaging, *Proceedings of the 36th International Conference on Software Engineering (ICSE 2014)*, New York, NY, USA, ACM, 2014, pp. 378–389.

[19] Utsugi, K., Obata, A., Sato, H., Katsura, T., Sagara, K., Maki, A., and Koizumi, H.: Development of an Optical Brain-machine Interface, *29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2007 (EMBS 2007)*, Aug. 2007, pp. 5338–5341.

[20] Yang, Y. and Raine, A.: Prefrontal structural and functional brain imaging findings in antisocial, violent, and psychopathic individuals: A meta-analysis, *Psychiatry Research: Neuroimaging*, Vol. 174, No. 2(2009), pp. 81–88.

[21] Zago, L., Pesenti, M., Mellet, E., Crivello, F., Mazoyer, B., and Tzourio-Mazoyer, N.: Neural Correlates of Simple and Complex Mental Calculation, *NeuroImage*, Vol. 13, No. 2(2001), pp. 314–327.

[22] 栗山進, 大平雅雄, 門田暁人, 松本健一: プログラム

理解度がコードレビュー達成度に及ぼす影響の分析, 電子情報通信学会技術研究報告. *SS,* ソフトウェアサイエンス, Vol. 104, No. 571(2005), pp. 17–22.
[23] 三輪和久, 杉江昇: 学習の初期段階における計算機プログラミングの動的過程, 人工知能学会誌, Vol. 7, No. 1(1992), pp. 138–148.
[24] 山口修平: 前頭葉と記憶, 高次脳機能研究, Vol. 27, No. 3(2007), pp. 222–230.
[25] 石黒誉久, 井垣宏, 中村匡秀, 門田暁人, 松本健一: 変数更新の回数と分散に基づくプログラムのメンタルシミュレーションコスト評価, 電子情報通信学会技術報告, ソフトウェアサイエンス研究会, Vol. SS2004-32, Nov. 2004, pp. 37–42.
[26] 村岸厳: 皮膚抵抗値によるソフトウェア開発者の負荷評価に関する研究, 修士論文, 奈良先端科学技術大学院大学, 1998.