

# Eliminating Stumbling Blocks Using Eye Tracking in Remote Programming Learning

Yuki TANI<sup>\*</sup>, Syoya SAKANAKA<sup>\*</sup>, Erina MAKIHARA<sup>\*\*</sup> and Keiko ONO<sup>\*\*</sup>

(Received October 14, 2022)

Due to the COVID-19 pandemic, the demand for remote programming learning has been increasing. Remote programming learning allows learners to adapt their skills and progress. However it tends to make communication more difficult. Therefore, remote programming learning is more difficult than face-to-face learning to resolve stumbling blocks in the debugging process for learners. This reaserch aims to support learners' self-learning and to help them resolve stumbling blocks in remote programming learning. We focused on the learner's gaze information as a method to eliminate stumbling blocks. In this paper, we propose a real-time visualization system of eye position to eliminate stumbling blocks for learners in remote programming. As a result, it is clear that the proposed system can estimate the approximate line of sight.

**Key words :** remote programming learning, gaze information, tips suggestion

**キーワード :** 遠隔プログラミング学習, 視線情報, ヒント提示

## 遠隔プログラミング学習における視線情報を用いた躓き解消手法の提案

谷 祐貴・坂中 翔矢・楨原 絵里奈・小野 景子

### 1. はじめに

IT 産業の拡大により IT 人材の需要が高まっている。しかし、IT 人材の需要が高まっていく一方、労働人口の減少が見込まれ、IT 人材の需要が供給を上回ると予想される。IT 人材の不足に伴い、プログラミング教育に高い関心が集まっており、プログラミングの学習環境は多様化する傾向がある。多様な学習環境のうち、COVID-19 の影響により遠隔プログラミング学習の需要が増加している。遠隔プログラミング学

習において、学習者は自身のスキルや進度に適した学習が可能になる。一方で、遠隔プログラミング学習はコミュニケーションが取りづらい問題を有する<sup>1)</sup>。対面学習と比較すると、学習者のデバッグ処理における躓きを解消することが困難である。

そこで本研究では学習者の自学自習を支援することで、遠隔プログラミング学習における躓き解消の支援を目指す。躓きを解消する手法として学習者の視線情報に注目した。視線情報からプログラミング初学者と熟練者の特徴が明らかになっている<sup>2)</sup>。そのため、プ

<sup>\*</sup> Graduate School of Science and Engineering, Doshisha University, Kyoto

Telephone:+81-774-65-6930, Fax:+81-774-65-6716, E-mail: tani.yuki@mikilab.doshisha.ac.jp

<sup>\*\*</sup> Faculty of Science and Engineering, Doshisha University, Kyoto

Telephone:+81-774-65-6930, Fax:+81-774-65-6716, E-mail: kono@mail.doshisha.ac.jp, emakihar@mail.doshisha.ac.jp

プログラミング熟練者の視線情報を躰いた学習者に提示することで躰きの解消につながる可能性がある。本研究では、遠隔プログラミング学習における学習者の躰き解消支援の前段階として、プログラミング熟練者のプログラム理解における視線情報が、学習者のプログラム理解に有効であるかを調査する。また、視線情報を学習者に提示するシステムを提案する。

## 2. 先行研究

### 2.1 視線情報に関する先行研究

藤本らは、視線の注視点が言葉では表現することが困難な着眼スキルの伝達に有効であるか調査した<sup>3)</sup>。具体的には、自動車運転免許を持たない被験者 30 名を 10 名ずつの 3 グループに分け、自動車運転時のシーンに対する「危機予知トレーニング」を行った。調査の結果、経験豊かな熟練ドライバーの視線位置から学習したグループは、他のグループと比較し、高い評価が得られた。したがって、視線情報は言葉では表現することが困難な事象を伝達する手段として有効な可能性がある。また、花房らは、ソースコードの構造と視線運動の関係性を調査した<sup>4)</sup>。具体的には、変数と演算子からなるソースコードと視線運動の関係について調査を行った。調査の結果、変数と演算子からなるソースコードと視線運動の読解手順が同じであることがわかった。したがって、ソースコードの構造と視線情報は関係があることが考えられる。

### 2.2 ヒントの提示に関する先行研究

Samiha らは、学習者のパフォーマンス向上のためにヒントを自動生成し、その有効性を調査した<sup>5)</sup>。具体的には、ヒントを提示したタイミングによって課題解決に変化があるのかについて調査を行った。評価実験の結果、ヒントを得た学習者は、課題の作成に関するパフォーマンスが向上した。また、鈴木らは、プログラミング初学者を対象にヒントの提示を行い、その有効性を調査した<sup>6)</sup>。学習者の学習進度に適応して自動生成したヒントがプログラミング初学者に有効かのアンケート調査の結果、8 割以上の被験者がヒントの提示は役立つと回答した。したがって、ヒントの提示は

プログラミング初学者にとって有効な可能性がある。

### 2.3 プログラム理解支援に関する先行研究

プログラム理解支援に関する研究は数多く存在する。鈴木らは、プログラムを解析して得られる情報を可視化する手法を提案した<sup>7)</sup>。具体的には、プログラムの構造およびプログラムを実行した際の各関数の関係性を可視化したツールを提案した。また、田中らは、プログラム理解のための付加注釈を提案した<sup>8)</sup>。具体的には、開発者が別の開発者に対してコードの意味などを説明するためのルールを作成したシステムを提案した。

## 3. 提案システム

一般的に視線計測器には専用のデバイスを用いて、視線情報を取得する。しかし、本研究では、オンライン環境での提案システム利用を想定しているため、デバイスに依存しない視線情報取得方法として WebGazer.js<sup>9)</sup>を採用した。WebGazer.js は、Web カメラを用いて視線情報を取得する視線アイトラッキングライブラリである。提案システムは、学習者の視線位置からリアルタイムでヒートマップの作成および注視した行の推定を行う。提案システムの全体構成を Fig. 1 に示す。学習者 A は課題を問題なく解ける人、学習者 B は課題を行う際に躰く人と仮定する。学習者 A と学習者 B は、まず機器の偏りを計測し正しい値に調整するためにキャリブレーションを行う。その後、学習者 A と学習者 B は提案システムが提示した課題に取り組む。本研究の課題はプログラミング学習での躰き解消を目指すため、ソースコードの読解を想定している。課題を行う際の学習者 A の視線位置を数フレームおきに Web サーバに送信する。その後、サーバで送信された視線位置をもとにヒートマップの作成および注視した行の推定を行う。そして、ヒートマップおよびサーバで推定した注視行を学習者 B に送信する。

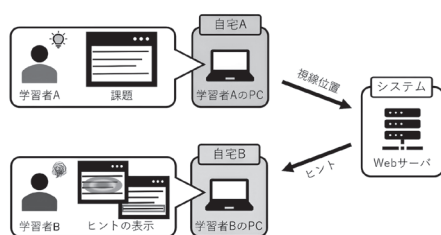


Fig. 1. Proposed System

#### 4. 提案システムの精度調査

##### 4.1 検証内容

本検証では、被験者が実際に注視した行（以後、注視行）と提案システムにより推定した行（以後、推定行）の比較の検証を行った。被験者は、C言語の基礎を履修済みの20代学生3名である。C言語で記述した15行のソースコードから指定した行を注視することを検証課題とした。また、注視時間は被験者の集中力を考慮し、10秒間注視することを指示した。

検証手順を以下に示す。

Step 1: キャリブレーションにより被験者の視線位置を初期化

Step 2: 検査官が注視行を指定し、被験者は指定された行を読解

Step 3: 提案システムを用いた視線位置から注視行の推定

なお、キャリブレーションでは9つの丸をそれぞれ5回クリックする作業を行った。そのため、被験者によってキャリブレーション時間は異なる。

##### 4.2 検証結果

被験者Aにおける指定行と推定行との整合率をFig. 2に示す。Fig. 2より、全ての行において整合率が0.5を下回っていることがわかる。しかし他の行と比較し、2行目、9行目、14行目、15行目は整合率が高いことがわかる。これはキャリブレーション時に使用したボタン位置と注視行の位置が類似する点が起因すると思われる。

被験者Bにおける指定行と推定行との整合率をFig. 3に示す。Fig. 2とFig. 3より、被験者Bは被験者Aと比較して整合率が高いことがわかる。被験者Bは他の被験者と比較し、キャリブレーションを長い時間行っていた。そのため、キャリブレーション時間が増加することで、視線位置を正確に推定する可能性が高い。各行における被験者3名の指定行と推定行との平均整合率をFig. 4に示す。Fig. 4より、すべての行において整合率が低いことがわかる。推定行を確認したところ注視行の前後にある行を推定しているため、整合率が低いことがわかった。よって、提案システムより視線位置からおおよその注視行を推定できる可能性が高い。また、3行目は他の行と比較して特に整合率が低いことがわかる。3行目のソースコードは1文字である。そのため、文字数の少ない行は整合率が低下する可能性がある。以上の結果より、学習者の注視行から行の推定を行うことは困難であるとする。

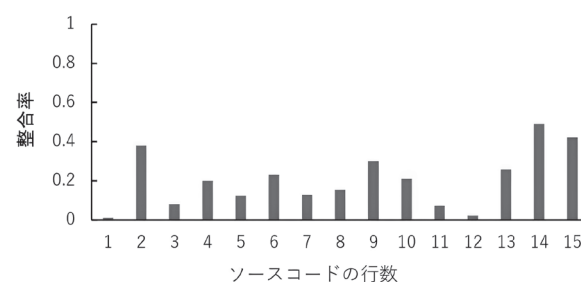


Fig. 2. Average consistency rate between specified and estimated rows for subject A

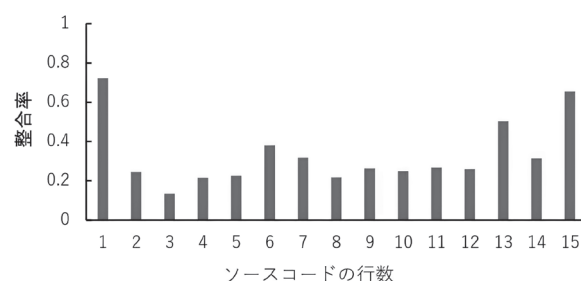


Fig. 3. Average consistency rate between specified and estimated rows for subject B

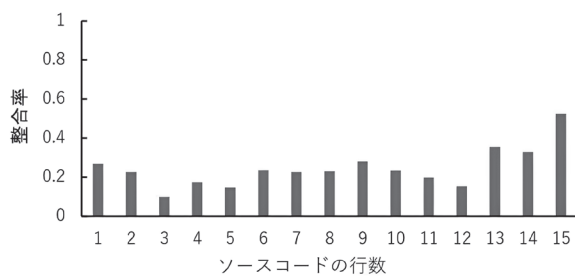


Fig. 4. Average consistency rate between specified and estimated rows

#### 4.3 考察および改善

被験者を1名追加し、実際に注視した行と注視行の前後にある行を提案システムが推定しているか検証した。各行における注視行と注視行前後の行を合わせた整合率を Fig. 5 に示す。Fig. 5 より、すべての行において整合率が高いことがわかる。また、すべての行において整合率が0.7を上回っていることがわかる。以上より、提案システムより視線位置からおおよその注視行を推定することが可能であると考えられる。

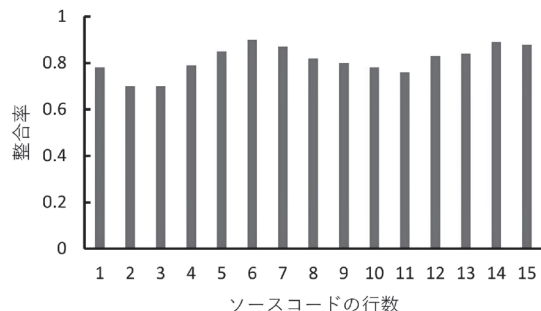


Fig. 5. Average Consistency Ratio for Multiple Lines

## 5. 視線情報の有効性調査

### 5.1 調査概要

本研究では、提案システムから取得した視線情報が置き解消に有効であるか調査した。視線情報は事前に用意した課題に対するプログラミング熟練者の視線運動から取得する。その後、取得した視線情報からヒントを作成する。そして、作成したヒントが置きに有効であるか否かに関して、被験者の意見をアンケートにて収集した。

### 5.2 ヒントの作成

事前に用意した課題内容を List. 1 に示す。List. 1 のソースコードは、先行研究<sup>10)</sup>においてプログラム理解のために利用されたソースコードを基に実験用に作成した。

List. 1. Source code

```

1  int a,b,c,d,e,f[]=new int[7];
2  a=0,b=2,c=4,d=5,e=3;
3  for(int i=0;i>20;i++){
4      a++;
5      if(a>10){
6          b=f.length;
7          break;
8      }
9  }
10 if(a<15)c=b+2;
11 else c=b+e;
12 d=b+c;
```

被験者に、List. 1 のソースコードを閲覧してもらったあと、以下に示す二つの質問を行った。

問1: 繰り返し処理は何回実装されているか。

問2: d の値を求めよ。

問1では、部分的な処理の読み取りに関する置きがある想定している。また、問2では、ソースコード全体の読み取りに関する置きがあると想定している。なお、本研究の置きとは、学習者がどの箇所を読み取れば良いか曖昧な状態と定義する。事前に用意した課題を開発経験のある被験者3名に解答してもらい、被験者3名の視線情報を取得した。そして、取得した視線情報からヒートマップと遷移グラフを作成した。各問いに対するヒートマップを Fig. 6 と Fig. 7 に示す。プログラミング熟練者の注視した箇所を容易に把握するために、ヒートマップを作成した、遷移グラフを Fig. 8 に示す。遷移グラフとは、学習者が注視した行の時系列情報を提示するグラフである。学習者の停留箇所および遷移順を可視化できる点から置き解消に有効であると考えため、遷移グラフを作成した。また、遷移グラフは各問に対して被験者3名の遷移グラフを作成した。

### 5.3 調査内容

本調査では、作成したヒントが各問いに対して有効なヒントであるか否かを5段階で評価を行った。被験



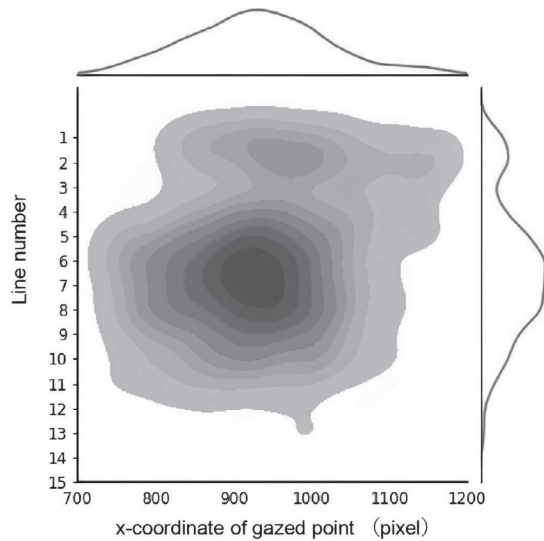


Fig. 6. Heatmap in Q1

者は、プログラミング歴が3年以上の20代大学生5名およびプログラミング歴が1年未満の20代大学生5名の計10名である。被験者は、作成した視線情報のヒントを利用し、作成したソースコードの課題を解く。その後、視線情報が有効なヒントであるか否かのアンケートに答える。

アンケート内容を以下に示す。

- Q1: ヒートマップの情報からヒントが得られたか  
 Q2: 被験者 A の遷移グラフからヒントが得られたか  
 Q3: 被験者 B の遷移グラフからヒントが得られたか  
 Q4: 被験者 C の遷移グラフからヒントが得られたか  
 Q5: ヒートマップと遷移グラフの両方からヒントが得られたか

#### 5.4 調査結果と考察

問1における被験者実験の結果を Fig. 9 に示す。Fig. 9 より、被験者10名の全ての問いに対して、評価平均が4を下回っていることがわかる。問1は、一部の処理における躓きがあると想定した問題である。よって、作成したヒントは、一部の処理に対する躓きに有効でないと考える。問2における被験者実験の結果を Fig. 10 に示す。Fig. 10 より、被験者10名の全ての問いに対して、評価平均が4を下回っていること

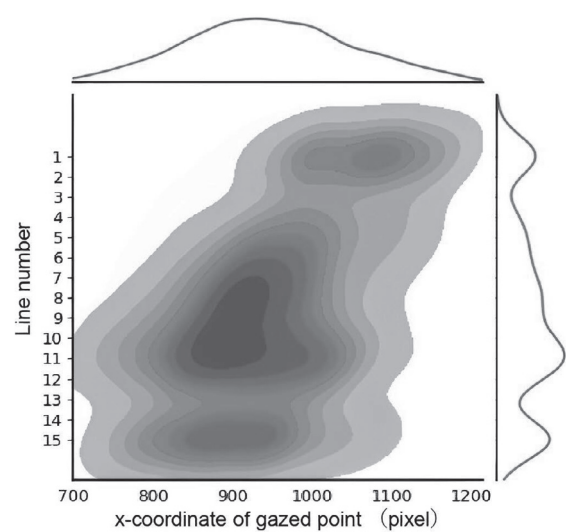


Fig. 7. Heatmap in Q2

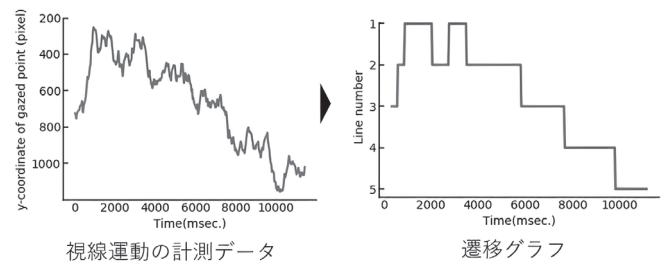


Fig. 8. Transition Graph

がわかる。問2はソースコード全体の読み取りに関する躓きがあると想定した問題である。よって、作成したヒントは、ソースコード全体の読み取りに対する躓きに有効でないと考える。以上の結果より、作成したヒントは、ソースコードの読み取りにおける躓きに有効でないと考える。

また、Fig. 9 と Fig. 10 より、プログラミング歴1年未満の評価とプログラミング歴3年以上の評価を比較すると、プログラミング歴1年未満の評価の方が低いことがわかる。よって、作成したヒントは、プログラミング初学者に対して、有効でないと考える。しかし、被験者が求めるヒントとして必要な要素を確認したところ作成したヒントが見つらいという意見が多く見受けられた。そのため、視線情報から作成したヒントを見やすくすることにより、高い評価が得られると考える。

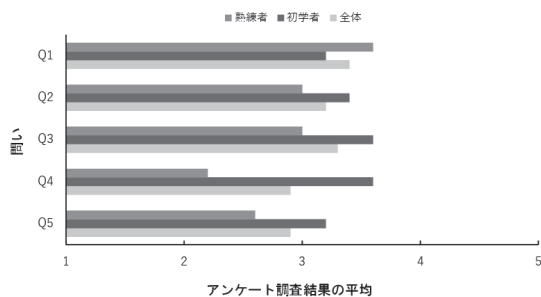


Fig. 9. Results of Questionnaire in Q1

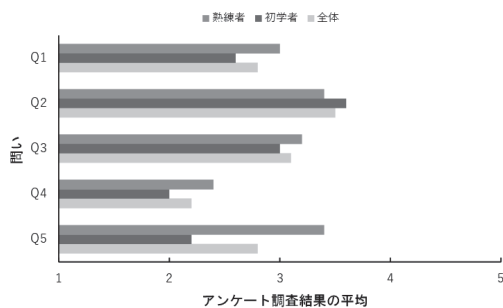


Fig. 10. Results of Questionnaire in Q2

## 6. 追加調査

### 6.1 調査内容

本調査は、視線情報の有効性調査（以後、前回調査）で得られた被験者の意見をもとに、改善したヒントが置きに有効であるか調査する。改善したヒントは、ソースコードを重ね合わせたヒートマップおよびプログラミング熟練者が注視した行の遷移を動画（以後、遷移動画）にしたものである。また、課題と視線情報は、前回調査と同様である。そして、改善したヒントが置きに有効であるか否かに関して、被験者の意見をアンケートにて収集した。各問いに対するヒートマップを Fig. 11 と Fig. 12 に示す。ヒートマップをソースコードと重ね合わせることで、注視した箇所を視覚的に把握することが容易であると考えた。

本調査は、前回調査と同様に、作成したヒントが各問いに対して有効なヒントであるか否かを 5 段階で評価を行った。被験者は、プログラミング歴が 3 年以上の 20 代大学生 2 名およびプログラミング歴が 1 年未

満の 20 代大学生 2 名の計 4 名である。被験者は、作成した視線情報のヒントを利用し、作成したソースコードの課題を解く。その後、視線情報が有効なヒントであるか否かのアンケートに答える。

アンケート内容を以下に示す。

Q1: ヒートマップの情報からヒントが得られたか

Q2: 被験者 A の遷移動画からヒントが得られたか

Q3: 被験者 B の遷移動画からヒントが得られたか

Q4: 被験者 C の遷移動画からヒントが得られたか

Q5: ヒートマップと遷移動画の両方からヒントが得られたか

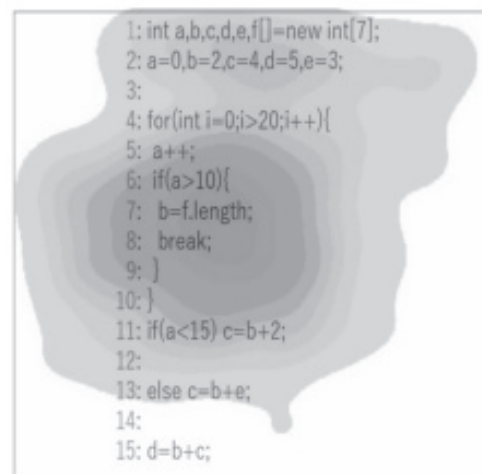


Fig. 11. Heatmap in Q1

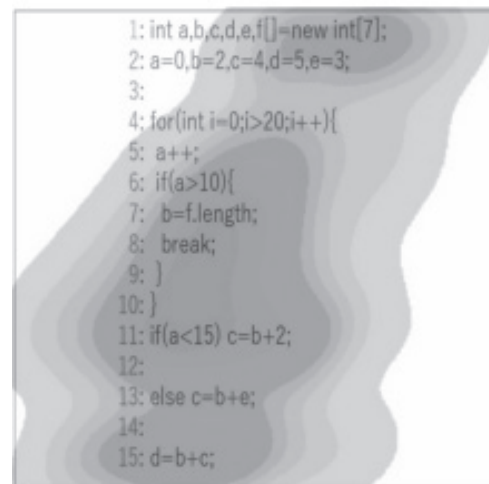


Fig. 12. : Heatmap in Q2

## 6.2 調査結果と考察

問 1 における被験者実験の結果を Fig. 13 に示す。Fig. 13 より、被験者 4 名に対する Q1 と Q5 の評価平均が高いことがわかる。問 1 は、一部の処理における躓きがあると想定した問題である。問 1 におけるヒント A は、躓きがある箇所を濃く表示していた。また、問 1 におけるヒント B は、躓きがある箇所を繰り返し遷移していることがわかった。そのため、一部の処理における躓きに対して、作成したヒントが有効であると考え。また、被験者 4 名に対する Q1 と Q5 を比較し、Q1 の方が評価が高いことがわかる。これは視線の遷移が 4 行目から 10 行目のみを繰り返し遷移していることに起因すると考える。そのため、一部の処理に対しては、ヒント A のみを提示することが有効であると考え。

問 2 における被験者実験の結果を Fig. 14 に示す。Fig. 14 より、被験者 4 名に対する Q1 と Q5 の評価平均が高いことがわかる。問 2 はソースコード全体の読み取りに関する躓きがあると想定した問題である。問 2 におけるヒント A は、問 2 を解くための必要な変数および条件分岐を濃く表示していた。また、問 2 におけるヒント B は、問 2 を解くための解答手順を表示していた。そのため、ソースコード全体の読み取りに関する躓きに対して、作成したヒントが有効であると考え。また、被験者 4 名に対する Q1 と Q5 を比較し、Q5 の方が評価が高いことがわかる。これはヒント A とヒント B の両方を表示することにより、解答に必要な変数と変数の使い方がわかることに起因すると考える。そのため、ソースコード全体の読み取りは、ヒント A とヒント B の両方を提示することが有効であると考え。

Fig. 13 と Fig. 14 より、初学者における評価と熟練者における評価を比較すると、初学者における評価の方が高いことがわかる。作成した課題がプログラミング初学者に適しているため、初学者における評価が高くなったと考える。

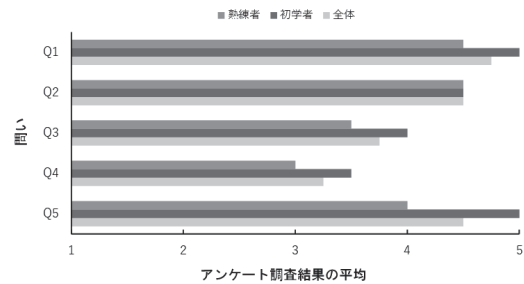


Fig. 13. Results of Questionnaire in Q1

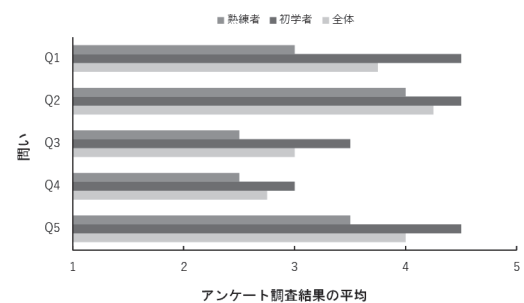


Fig. 14. Results of Questionnaire in Q2

## 7. おわりに

本論文では、遠隔プログラミング学習における学習者の躓き解消を行うために視線位置をリアルタイムで可視化したシステムを提案した。また、学習者の視線位置から注視行の推定が可能かを検証した。結果として、視線位置から行の推定は容易でないことがわかった。しかし、キャリブレーション時間を増加させることで精度が向上する可能性を示した。また、複数行の精度調査を行った結果、注視行の前後にある行を推定することは可能であることがわかった。そのため、提案システムを用いることでおおよその注視行を推定することは可能であることが明らかになった。また、視線情報がソースコードの読み取りにおける躓きに有効であるかの調査を行った。結果として、視線情報は見せ方によって評価が異なることがわかった。しかし、視線情報は、見せ方を工夫することにより、ソースコードの読み取りにおける躓きに有効である可能性が高いことが明らかになった。加えて、本調査では、プログ

ラミング歴によらず、ソースコードの読み取りにおける躓きに有効である可能性を示した。本研究の取り組みにより、視線情報が遠隔プログラミング学習の躓き解消に有効かの基礎となる知見を得た。提案システムは、ブラウザを開くことで使用することができるため、対面学習での使用も望める。そのため、遠隔プログラミング学習のみならず、対面でのプログラミング学習の躓き解消にもつながると考える。

### 参考文献

- 1) 高原利幸, 宮里心一, “オンライン講義と対面講義における学生の意識比較”, 工学教育研究, **29**, 51-57 (2021).
- 2) 金藤光平, 石原真紀夫, “プログラミングの熟練者と初学者のコード読解時における視線情報に関する一考察”, 電気関係学会九州支部連合大会講演論文集, 410-410 (2019).
- 3) 藤本武司, 砂山渡, 山口智浩, 谷内田正彦, “視線行動の可視化による着眼スキル伝達支援”, 人工知能学会論文誌, **19**[3], 174-183 (2004).
- 4) 花房亮, 松本慎平, 平嶋宗, 林雄介, “初学者のソースコード読解における視線運動とデータ依存構造の関係分析”, 先進的学習科学と工学研究会, 192-203 (2018).
- 5) S. Marwan, J. J. Williams, T. Pice, “An Evaluation of the Impact of Automatic Programming Hints on Performance and Learning”, *International Computing Education Research Workshop*, 61-70 (2019).
- 6) 鈴木孝幸, 納富一宏, “個人適応型ヒント提示・誘導システムを用いたプログラミング演習授業の実践”, 情報科学技術フォーラム, 337-338 (2018).
- 7) 鈴木宏紀, 山本普一郎, 阿草清滋, “プログラム実行情報の視覚化による理解支援ツール”, 情報処理学会研究報告, **1998**[64], 77-84 (1998).
- 8) 田中昌弘, 石尾隆, 井上克郎, “プログラム理解のための付加注釈 DocumentTag の提案”, 情報処理学会研究報告, **2009**[31], 201-208 (2009).
- 9) A. Papoutsaki, P. Sangkloy, J. Laskey, N. Daskalova, J. Huang, and J. Hays, “WebGazer: Scalable Webcam Eye Tracking Using User Interactions”, *International Joint Conferences on Artificial Intelligence Organization*, 3839-3845 (2016).
- 10) 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, **104**[466], 37-42 (2004).