

自己組織化マップを用いた視線情報の分析による特徴抽出 A Study of Feature Extraction by Analyzing Gaze Information Using Self-Organizing Map

酒井 泰地[†]
Taichi Sakai

浦野 昌一[†]
Shoichi Urano

1. はじめに

情報活用能力が必要となった近年、学校の情報科目ではプログラミングをはじめ、データ活用などの専門的な学習をするようになる。しかし、教育者の育成が十分でないことから教育者を支援するシステムの重要性は高まっている。

従来の教育者を支援するシステムは新たな教材の提案や解答結果を分析するものが多い^[1]。しかし、実際の教育現場では教育者が解答の過程を見て個人に合わせた指導を行っている。そこで本研究では問題の解答結果のみではなく、解答の過程に着目したプログラミング学習の教育改善を目的とする。

これまでに、教育支援を目的とした視線情報を用いた研究として、重要でない箇所に視線が集中した際に重要な箇所に色をつけてヒントを出すことで学習を支援する研究が報告されている^[2]。本稿では、問題解答の際には解答に重要でない箇所にも視線が集中することがあることを考慮し、解答過程全体の特徴抽出を行う。プログラミング問題解答時の視線情報を計測し、視線情報に分析手法を適用して分類および分析することで効果的な学習支援システムの構築を目指す。筆者らはこれまでに視線情報のクラスタリングに K-medoids クラスタリングが有効であることを確認した^[3]。そこで、今回は視線情報に K-medoids クラスタリングと自己組織化マップを用いて解答過程を可視化し、特徴を抽出することで、問題の正答者と誤答者の特徴比較を行う。

2. アイトラッキング

プログラミング問題解答時の視線計測には、Tobii eye tracker nano pro を使用した。視線情報測定の様子を図 1 に示す。この計測装置は、画面下部に取り付けて赤外線を照射し、カメラ画像処理と合わせて被験者の注視点点を計測するセンサーである。サンプリングレートは 60Hz である。

Tobii eye tracker nano pro は計測前に被験者に画面の特定箇所を注視させることによりキャリブレーションを行う。これにより、画面サイズや画面からの距離に合わせることに加え、被験者の注視点と計測点のずれを調整する。計測

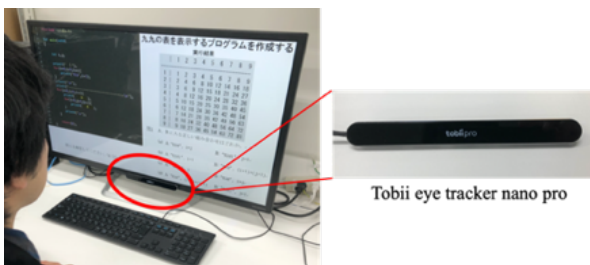


図 1 アイトラッキングのイメージ

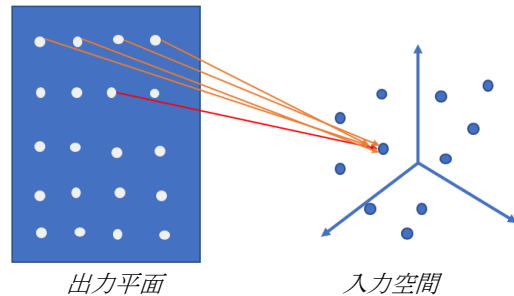


図 2 自己組織化マップのイメージ図

時にはキャリブレーションにより生成された角膜の赤外線反射パターンと画像処理によって注視点点を計算している。

3. 分析手法

視線情報の分類手法として K-medoids クラスタリングを適用する。また、分類したデータをマッピングし分析を行うために自己組織化マップを用いる。

3.1 K-medoids クラスタリング

クラスタリングは分類対象のデータ群から類似するデータが同じクラスになるように分類するデータ解析手法の一つであり、手法によって特徴が違うため分類も変化する。K-medoids クラスタリングは、最も一般的な K-means クラスタリングの重心計算の代わりに、クラスタ内の全ての点との距離の総和が最小になる点を中心とする手法である。K-means クラスタリングと比べて外れ値の影響を受けにくいという特徴がある。

3.2 自己組織化マップ

自己組織化マップは高次元観測データを低次元に写像する教師なしニューラルネットワークの一種である。自己組織化マップの概念図を図 2 に示す。出力平面のノードと入力空間の一点を指す参照ベクトルによって構成される。各入力データに対し、最も近い参照ベクトルを持つノードを勝者ノードとし、勝者ノードと近傍ノードの参照ベクトルを学習によって更新することで、性質の近いものが隣り合うようにマッピングする。アルゴリズムを以下に示す。

- I. 出力平面、参照ベクトルを初期化する
- II. 各入力ノードに対し、勝者ノードを求める
- III. 勝者ノードと近傍ノードの距離から学習量を求める
- IV. 学習量に応じてそれぞれのノードを入力に近づける
- V. 全入力ノードに対し II ~ IV を繰り返し、出力平面を更新する

4. 実験

実験手順を図 3 に示す。被験者ごとにキャリブレーションを行い、注視点の調整を行ったのち、プログラミング問題を画面表示し解答する。解答の際計測された視線情報に K-medoids クラスタリングおよび自己組織化マップを適用

[†] 明治大学 Meiji University

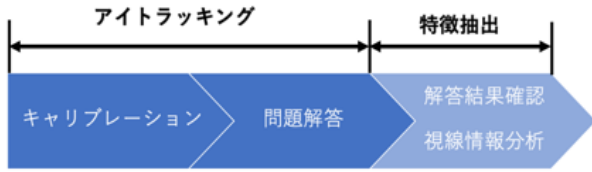


図 3 実験フロー

することで、出力画像から正答者と誤答者の解答過程を確認し、特徴の変化を比較する。

4.1 実験条件

視線情報計測時と視線情報分析時の条件を以下に示す。

4.1.1 アイトラッキング時の条件

C 言語プログラミング経験があり、学力にばらつきのある大学生 10 人を被験者とする。解像度 1920×1080 ピクセルの画面で計測を行い、注視点はその座標点で出力する。出題する問題は簡単な繰り返し文などの処理内容を問う穴埋め形式の問題とし、プログラムが正しく動作するように 4 つの選択肢から解答する。ソースコード、実行結果、問題を同時に画面に表示することで視線情報から解答手順全体を把握できるようにしている。

必要に応じて手元にメモを取りながら解答する。制限時間を設け、時間内に解答できなかった場合は誤答とする。

4.1.2 特徴抽出時の条件

特徴抽出は解答過程を自己組織化マップにより二次元画像に出力し、K-medoids クラスタリングにより分類されたクラスをラベル付けしている。また、実験時間を等間隔で 10 フェーズに分割し、フェーズごとに色付けをしている。K-medoids クラスタリングを用いた際に設定した条件は以下の通りである。

クラス数: 要素の数

初期値: 要素の座標範囲の中心付近

クラス数を決める要素を図 4 に示す。今回それぞれのクラスが意味する内容は以下の通りとする。

クラス 1: for 文 1 クラス 4: 実行結果

クラス 2: for 文 2 クラス 5: 選択肢

クラス 3: 問題文 クラス 6: 手元のメモ

自己組織化マップの条件は以下の通りとする。

入力データ: 視線座標, 時間

ノード数: 480 近傍関数: ガウス関数

初期更新半径: 10 初期化: PCA

最終更新半径: 1 学習回数: 1000

4.2 実験結果

誤答者の解答過程の分析結果例を図 5 に示す。また、正

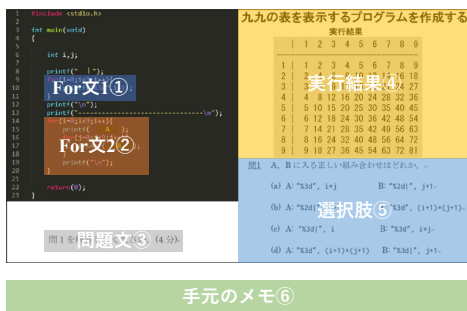


図 4 クラスタを決める要素

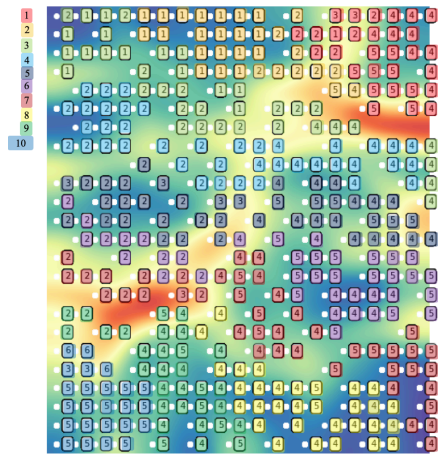


図 5 誤答者の視線情報分析結果例

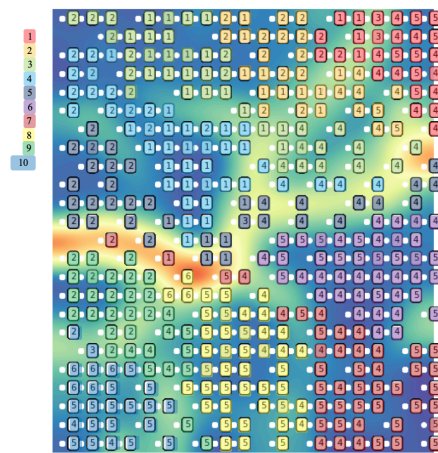


図 6 正答者の視線情報分析結果例

答者の解答結果の分析結果例を図 6 に示す。画像を色ごとに見ることで被験者がいつどの領域を見ているかを把握することができる。また、誤答者と正答者の画像を比較すると正答者の方がフェーズごとに見ている領域が定まっていることがわかる。これは、誤答者は問題の解釈と選択肢の吟味を同時に行なっているのに対し、正答者は問題を解釈したうえで選択肢を考え解答したためであると考えられる。

5. おわりに

本稿では、プログラミング問題解答時の視線情報に K-medoids クラスタリングおよび自己組織化マップを適用してマッピングすることで問題解答過程を可視化し、正答者と誤答者の比較を行った。

今後は、より細かい時間変化を考慮して解答過程の特徴抽出を行うとともに、学習者に対するフィードバックを検討していきたいと考えている。

参考文献

- [1] 清水 康敬, 堀田 龍也, 中川 一史, 森本 容介, 山本 朋弘, “教員の ICT 活用指導力を向上させる研修システムの開発”, 日本教育工学会論文誌, Vol.34, No.2 (2010).
- [2] Jerry Chih-Yuan Sun, Kelly Yi-Chuan Hsu, “A smart eye-tracking feedback scaffolding approach to improving student learning self-efficacy and performance in a C programming course”, Computers in Human Behavior, Vol.95, pp.66-72, (2019).
- [3] 酒井 泰地, 浦野 昌一, “アイトラッカーを用いたプログラミング学習改善の検討”, 人工知能学会全国大会, 4J2-GS-2-04(2020).