

写経学習における習熟度と視線情報の関係分析

田中 竜馬[†] 尾崎 知伸[†]

[†] 日本大学 文理学部

1. はじめに

プログラミング学習では、一般に、教員や教材が提示したサンプルプログラムを学習者が自らの環境で模倣して入力・実行・結果の確認を行う。この一連の手順を含む学習方法は写経学習¹⁾と呼ばれるが、この写経学習を学術的な観点で扱った研究は必ずしも多くない。

本研究では、同じ状況下の写経学習における学習者間での学習効果のばらつきに焦点を当て、学習効果のばらつきには学習者の思考過程が要因の1つであると仮定し、プログラミングスキルの異なる2つのグループに対して写経学習中の視線情報を計測する比較実験を行った。

2. 実験

本章では、写経学習中の視線情報を計測する実験環境と、写経学習の具体的な手順について述べる。

なお、写経学習の具体的な内容や方式は教材や指導者によって多種多様であるが、ここで扱う写経学習は「提示されたサンプルプログラムを学習者が自らの実行環境にて提示された内容の通りに入力する学習方法」とし、テキストによる解説や教師による指導などを一切含まないものとする。理由としては、本研究の目的は学習方法の1つとしての写経学習の性質を明らかにする要素を示すことであり、写経学習が含む様々な学習方法を比較することではないためである。

2.1 実験環境と被験者

視線追跡ツールには WebGazer²⁾ を使用し、ウェブブラウザ上にサンプルプログラムを表示する領域と、実際にプログラムの入力を行うエディタを表示する領域を配置した。これにより、サンプルプログラムを読んでいる時とエディタにプログラムを入力している時の視線追跡を行うことができる。

今回の実験では、情報科に属する4年生8名の被験者を3年次までのプログラミング科目の成績から初級者グループ(A1～A4)と上級者グループ(B1～B4)にそれぞれ4人ずつにグループ分けを行った。

また、実験に使用するサンプルプログラムは、写経学習用の書籍¹⁾に掲載されているC言語のサンプルプログラムの2種類と、同じく同書籍に掲載されているサンプルプログラムの内容はそのままに、著者によりPythonで書き換えたプログラムの2種類の計4種類である。

2.2 実験手順

実験の流れと学習内容を以下に示す。

- (1) 「キャリブレーション」 キャリブレーションを通じ、

```
#include<stdio.h>
int maxdata(int d[], int num)
{
    int i;
    int max = d[0];
    for (i = 1; i < num; i++) {
        if (d[i] > max) {
            max = d[i];
        }
    }
    return(max);
}

int main(void)
{
    int data[5] = {1, 25, 19, 53, 9};
    int maximum;
    const int NUM = 5;
    maximum = maxdata(data, NUM);
    printf("max = %d", maximum);
    return 0;
}
```

図1 サンプルプログラムの例 (C言語)

被験者にディスプレイのどの辺りまで視線追跡が行われているか、どの程度顔を動かしてよいのかを把握させる。

- (2) 「英文の複写」 この段階で初めてサンプルプログラム画面部分とエディタ画面部分を表示する。ここでは、英文の複写とプログラミングの写経学習を比較することで、英文を書き写すという単なる複写では現れなかったプログラミングの写経学習における特徴やグループ間の差を取り出す。
- (3) 「C言語での写経学習」 サンプルプログラムとして、配列と関数をテーマとする2種類のプログラムを提示する。なお、プログラム自体は書籍¹⁾から引用した。サンプルプログラムの例を図1に示す。
- (4) 「Pythonでの写経学習」 Pythonの2種類においても、配列と関数をテーマとしたサンプルプログラムを引用した。なおC言語での学習時とは異なる内容のサンプルプログラムを利用した。

3. 視線の分析

前述した実験により得られた視線情報に行った2つの前処理、及び前処理後のデータに利用する特徴量とその特徴量から考察される内容を述べる。

3.1 注視点データと注視点グラフ

実験から得られた視線情報に Buscher らの手法³⁾を用いて fixation-saccade 判定を行う。fixation-saccade 判定を行った軌跡データをここでは注視点データと呼ぶ。fixation-saccade

An Experimental Analysis of the Relationship between Programming Proficiency and Eye Movements in Syakyo-Style Learning by Ryoma Tanaka and Tomonobu Ozaki (College of Humanities and Sciences, Nihon University)

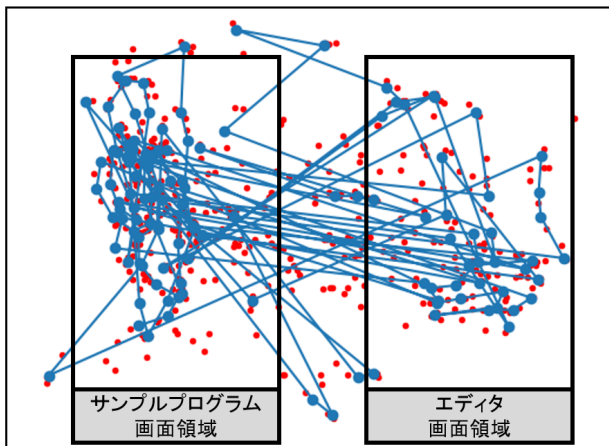


図2 fixation-saccade 判定を行った視線データ

判定の結果の例を図2に示す。

次に、注視点データから時系列の情報を除いたグラフを取り出す。時系列を無視し各注視点の位置と繋がりに注目するため、2つ目の前処理として、互いの距離が一定距離 $S = 50$ 以内に位置する注視点の一つの注視点とし、無向グラフを作成する。このグラフを注視点グラフと呼ぶ。

3.2 特徴量

注視点データ及び注視点グラフから、それぞれ以下の特徴量を抽出する。

注視点データにおける特徴量

- (1) 注視点の個数
- (2) 画面別での注視点の個数
- (3) 注視点間の総距離

注視点グラフにおける特徴量

- (1) ノードの個数
- (2) ノード間の平均距離
- (3) ノードの平均次数
- (4) ネットワーク中心性 (次数中心性・離心中心性・媒介中心性・固有ベクトル中心性・PageRank)

4. 結果と考察

注視点データ及び注視点グラフに対する特徴量抽出結果を表1に示す。

注視点グラフと比べて注視点データには、グループ間で視線情報の傾向に大きな差が確認できた。画面全体での注視点の個数と注視点時間には両グループとも被験者やプログラムによってばらつきが多いが、画面別の注視点の個数・注視点間の総距離においては、上級者グループの方がサンプルプログラム側の画面を広く長く注視し、色々なタイミングで入力箇所以外を注視したという結果となった。これは、写経学習中に書き写すプログラムを覚える時間以外に、より長い時間サンプルプログラムを全体的に注視していたからであると考えられる。

注視点グラフに関しては、注視点データほど多くの被験者において共通して見られた傾向は少ない結果となったが、

表1 注視点データにおける特徴量

特徴量	A1	A2	A3	A4	B1	B2	B3	B4
画面全体での注視点の個数	211	178	198	252	120	140	132	233
エディタ領域での注視点の個数	131	86	104	146	47	53	40	91
注視点間の総距離 (cm)	320	269	247	311	253	188	287	341
ノードの個数	77	60	66	74	62	65	54	84
ノード間の平均距離 (cm)	142	197	170	155	246	116	169	88
ノードの平均次数	3.6	4.8	2.1	2.6	5.4	5.1	4.6	3.0

ノードの平均次数・次数中心性のようなノードの次数が影響する2つの特徴については、上級者グループの方が平均次数が高くサンプルプログラム側の画面の中心性が高い傾向が確認された。このようになった理由は、注視点データと同じくサンプルプログラムを注視していたために、サンプルプログラム側の画面上で視線運動が多く行われ、次数と次数中心性が高くなったと推測される。

また、媒介中心性や固有ベクトル中心性には両グループ共にばらつきが多く目立った傾向が見られなかったが、写経学習自体の性質からしてサンプルプログラムとエディタを交互に見る回数がどちらのグループにおいても当然ながらとても大きくなるために、それぞれのグループの特徴が埋もれてしまったと考えられる。

以上の結果から、傾向として読み取れることは、上級者グループは初級者グループと比べて、サンプルプログラムを注視しており、かつ、より多くの視線運動を行っていることであると考察される。

5. おわりに

本研究では、学習者のプログラミングスキルと写経学習での視線データの関係分析を行った。今回の結果だけからでは「プログラミングスキルの高い者とそうでない者では、写経学習中の思考過程が異なり、その思考過程がプログラミングスキルの向上に繋がっている」という仮説の証明には至ってはいないが、「プログラミングスキルが高い者とそうでない者の写経学習中の視線運動には特徴的な差がある」ことを示し、写経学習における学習効果の差が生まれる要因の1つが写経学習中の思考過程にあることの可能性を示唆していると考えられる。

今後の課題としては、さらに教材やエラーパターンなどの写経学習に関わる様々な要素を考慮することが考えられる。

参考文献

- 1) 岡本 雅子, 喜多 一, 藤岡 健史, 吉川 直人 (2012) 写経型学習による C 言語プログラミングワークブック 共立出版
- 2) A. Papoutsaki, P. Sangkloy, J. Laskey, N. Daskalova, J. Huang and J. Hays: WebGazer: Scalable Webcam Eye Tracking Using User Interactions, *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, pp.3839-3845 (2016)
- 3) G. Buscher, A. Dengel and L. vanElst: Eye movements as implicit relevance feedback, *Computer Human Interaction '08 extended abstracts on Human factors in computing systems*, pp.2991-2996 (2008)