

組込みシステム開発のデバッグにおける視線と熟練度の関係分析

馬場 建
Takeru BABA

1 はじめに

近年、業務効率改善を目的とする IoT 市場規模の拡大に伴い、IoT 技術者の需要が増加すると予想されている。IoT の開発は、組込みシステム開発を基礎技術として、ネットワークやセンシング、データの分散処理などの応用技術が必要であり、技術の習得が容易でない。IoT の基礎技術である組込みシステム開発の効率化は、今後のソフトウェア工学の課題になると考えられる。

組込みシステム開発は、一般的なソフトウェア（以降、SW）開発に加えてハードウェア（以降、HW）の要素が増える。組込みシステム開発における、システムが正常に動かない原因、すなわち誤りの箇所は、SW だけでなくマイコンや回路などの HW にも発生しうる。多くの場合、HW と SW の誤りは見分けがつかず、システムの誤り箇所を推定し、修正することは初学者にとって容易でない。熟練者の持つ組込みシステム開発のデバッグ特有のコツを分析することで、作業効率向上のための教材の拡充が可能である。したがって本研究では、組込みシステム開発のデバッグ作業における視線動作を定量的に分析し、熟練者と初学者のデバッグ過程の違いを明らかにすることを目的とする。

2 アイトラッカーによる視線情報の分析

2.1 アイトラッカーによる定量分析

視線には、思考プロセスや能力によって動きが変化するという特徴がある。アイトラッカーを用いると、装着者が見た対象へ向かう視線が停留した座標を取得できる。停留座標に関する分析手法は、統計的分析、時系列分析、認知プロセスの推定の 3 種類ある¹⁾。統計的分析は注目座標に関する時間的な分布を、時系列分析は注目の順序関係を調べる方法である。そして認知プロセスの推定は、PC の作業ログと視線の遷移を関連付けることで、人の認識過程を推定する方法である。

視線の停留座標と映像上の物体領域情報を組み合わせると、視線情報から分析できる事項が多くなる。例えば、製品の検品作業においてどの部分にどの頻度で着目するのかという情報は、注目物体を分析することで可能となる。

2.2 アイトラッカーにおける従来の注目物体検出手法

ウェアラブルアイトラッカーを用いた注目物体検出は、人間が手作業で行うことが主流であった。しかしアイトラッカーの精度の低さやキャリブレーションのずれから、人間の定性的な判断が注目物体の推定に影響する可能性があり、それを防ぐためには、コンピュータにより機械的に位置の推定が行われる必要がある。

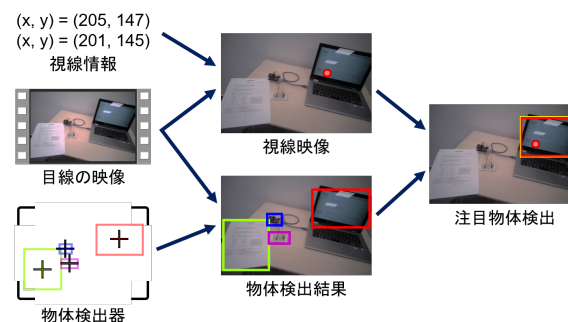


Fig.1 注目物体自動検出のイメージ

2.3 注目物体検出の提案手法

本研究では、映像上の物体位置をコンピュータに推定させるため、YOLO v3 という機械学習手法で物体検出器を学習させた²⁾。3.2 節で示すような、被験者が見る可能性がある四つの要素に関して、物体位置を推定できるように物体検出器を作成した。

学習させた物体検出器を用いて、被験者が注目した物体の検出を行う。物体検出器と視線情報を用いて、下記のような手順で注目物体検出を行なった。

- 手順 1 視線情報と目線の映像、物体検出器を入力
- 手順 2 視線の停留が生じた映像のフレームに移動
- 手順 3 物体検出器にそのフレームの画像を入力
- 手順 4 物体検出器から物体位置を長方形領域として取得
- 手順 5 視線の停留座標が物体領域内であれば注目物体として認識

3 組込みシステム開発に関する視線分析実験

3.1 実験概要

本実験の目的は、組込みシステム開発において、視線情報から数学的モデルを生成することで、熟練度の差の要因を顕在化することである。差を分析するために被験者に課題を与え、視線情報とデバッグの操作ログを記録する。

3.2 実験条件

被験者は、組込みシステム開発経験がある大学生の 7 名である。本研究では経験により得られる組込みシステム開発に固有の特徴を明らかにしたいため、経験年月数が長い被験者を熟練者とする。したがって、組込みシステム開発経験が 1 年以上である 3 名を「熟練者」、1 年未満である 4 名を「初学者」と定義する。実験のために構築した組込みシステム開発の環境の要素を下記に述べる。

- ディスプレイ：ソースコードと文字出力
- 問題用紙：回路や関数、コンパイル方法などを記述

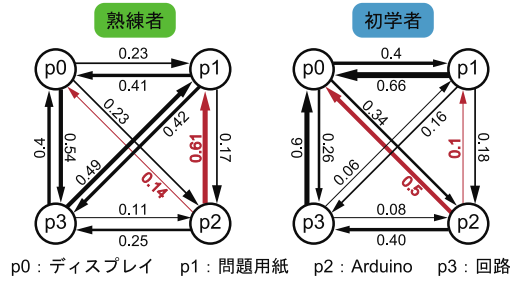


Fig.2 実験の終盤における注目物体のマルコフモデル

- Arduino : 電圧を制御するマイコン
- 回路 : デジタル入出力に関する回路

被験者に与える課題は、スイッチの押下時に一定間隔でLEDが点滅し、押下時以外はLEDが消灯するシステムの構築である。被験者に課題として与えるシステムには、ディスプレイと回路に二つずつ誤りが含まれている。

3.3 実験手法

熟練度の違いによって生まれる差を分析するために、被験者に組込みシステムのデバッグを行う課題を与えた。実験開始時には、システムの完成形を見せながら課題内容やコンパイル方法の説明を行った。説明の後、被験者はシステムのデバッグ作業を行った。実験中に取得する情報は、アイトラッカーによる視線情報、ソースコード編集履歴などである。実験の終了後、組込みシステム開発の経験歴に関するアンケートを実施した。

3.4 実験結果

3.4.1 実験時間の部分的な傾向に着目した分析

被験者実験で取得した情報から熟練度の差を起こす要因を分析するために、注目物体の時間的推移において熟練度ごとにデバッグの方針が現れるかを調査した。そこで、実験結果を視線の停留回数で三等分に分割して分析することで、デバッグ箇所の偏りが存在し得るか検証した。

熟練者と初学者の注目物体のマルコフモデルを三つの段階で比較すると、終盤のモデルに顕著な違いがみられた。熟練者と初学者の、終盤における注目物体のマルコフモデルを Fig. 2 に示す。各遷移確率に関して有意水準 5% の Welch の t 検定をした結果、Fig. 2 の確率値が太字で示されている遷移に関して有意差があった。

問題用紙からディスプレイと回路に向かう遷移確率に着目すると、熟練者は両方の遷移確率が近い値を示したことから、終盤において熟練者はソースコードと回路を修正していると考えられる。また、ディスプレイと回路間の双方向の遷移確率が近い値を示したことから、終盤において熟練者はソースコードと回路に誤りがある可能性を等しく疑うことが示唆される。終盤において初学者は、問題用紙からディスプレイへの遷移確率が高い値であり、終盤において初学者はソースコードを修正していると考えられる。

3.4.2 時系列データの数学的な分割

熟練者の修正方針を把握できた場合、組込みシステム開発における効率的なデバッグの手順を明らかにできる可

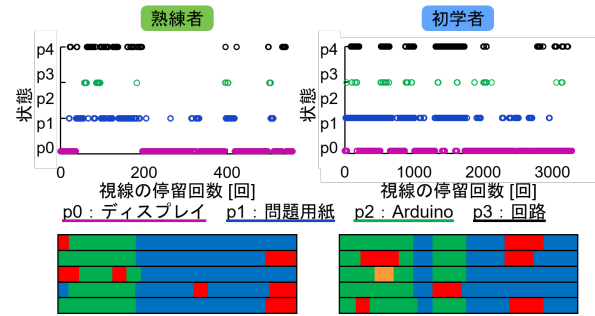


Fig.3 GP-HSMM による時系列データの分割

能性がある。したがって、熟練度ごとに修正方針が数学的な分割手法で適切に推定できるかを調査した。時系列データの分割をするために、GP-HSMM (Gaussian Process - Hidden Semi Markov Model) というガウス過程と隠れセミマルコフモデルを組み合わせた手法を用いた。

実験時間について 5 回ずつ分割を試行した際の、ある 2 名の被験者の結果を Fig. 3 に示す。時系列データを分割した結果、熟練者は Fig. 3 (左下) の 3 つのクラス、初学者は Fig. 3 (右下) の 4 つのクラスに分割された。各クラスの意味を調べると、熟練者は緑が「回路と問題用紙」に注目、赤が「ディスプレイと問題用紙と回路」に注目、青が「ディスプレイ」のみに注目する状態だった。初学者は緑と青が熟練者と同じ状態で、赤が「ディスプレイと問題用紙」に注目、黄が「回路と Arduino」に注目する状態だった。

熟練者は、赤のクラスが人間にとって曖昧な状態であり、修正方針を断定できない結果となった。初学者は、修正箇所注目した場合、緑と黄が回路、赤と青がソースコードを修正している状態であり、初学者は前半に回路、後半にソースコードを修正することが明らかになった。

4 おわりに

本研究では、組込みシステム開発のデバッグにおける視線動作について実験を行い、熟練度ごとに視線動作の違いがあるか分析を行った。作業終盤においてシステムの動作確認をする際、熟練者はソースコードと回路に誤りがある可能性を等しく疑う傾向があり、初学者はソースコードに誤りがある可能性を疑う傾向があることを確認した。また、時系列データの数学的な分割手法を用いたところ、熟練者は 3 つのフェーズがある可能性が示唆され、初学者のデバッグにおいて 2 つのフェーズの存在が確認できた。

現段階の分析方法では、熟練者のデバッグのフェーズは人が理解できる内容でない。したがって、今後の展望としては異なる時系列データの分割手法を検証する。また、本稿までに判明した熟練者のコツを、初学者に教示する影響を調査する。

参考文献

- 1) 大野健彦, 視線から何がわかるか, 認知科学, Vol. 9, No. 4, pp. 565-579 (2002).
- 2) Redmon, J. and Farhadi, A.: YOLOv3: An Incremental Improvement (2018).