Vol.2020-MPS-129 No.10
2020/7/27

組込みシステム開発のデバッグにおける視線情報の分節化による修正方針の分析

馬場 建[1,a)]　槇原 絵里奈[2,b)]　米田 浩崇[3]　清川 清[1]　小野 景子[2]

概要：組込みシステム開発では，ハードウェアとソフトウェアの両状態を考慮する必要があり，初学者にとってデバッグを行なうことは難しい．本研究では，無意識的な知識が反映される視線に着目し，熟練者がバグを効率的に修正する方針を視線分析で明らかにできるかを調査する．そのために，熟練者と初学者の組込みシステムデバッグ時の視線情報を分析し，熟練者の無意識的な知識と，初学者が苦手とする点を調査した．分析は注目物体の時系列データに対して，GP-HSMM という高精度な教師なしの分節化手法を用いることで行なった．分析の結果，初学者はデバッグの修正方針を持たず，システムの確認方法を知らないことがわかった．熟練者は序盤に回路，中盤にソースコード，終盤に回路とソースコードの両者を確認する傾向があることが明らかになった．

Gaze Analysis of Modification Policy in Debugging an Embedded System

Baba Takeru[1,a)]　Makihara Erina[2,b)]　Yoneda Hirotaka[3]　Kiyokawa Kiyoshi[1]　Ono Keiko[2]

1. 序論

Internet of Things（IoT）市場の拡大につれて，IoT 技術者の需要が高まっている．IoT の基礎技術である組込みシステムは，ソフトウェア（SW）だけでなく，マイコンや回路などのハードウェア（HW）の開発が必要となり，システムの誤りが生じる領域が増加する．SW 開発と異なり，HW 開発では回路の状態をもとにエラー出力をすることが容易でないため，システムの状態を調べるために HW と SW を見比べてバグの位置を推測する必要がある．よって，組込みシステム開発において，デバッグ能力が視線の振る舞いに反映されている可能性が高い．

本稿では，熟練者と初学者におけるデバッグ中の視線動作を分析することで，熟練者のコツと，初学者が躓きやすい点を調査する．実験で収集した視線情報を時系列分析することで，開発者が序盤から終盤にかけて何に注目していたかを明らかにできる．時系列データの分節化のために，多次元データを高速に高精度で分割できる Gaussian process-hidden semi-Markov model（GP-HSMM）という中村らの手法を用いる [1]．提案手法により熟練度ごとの修正方針を明らかにする．

[1] 奈良先端科学技術大学院大学
Nara Institute of Science and Technology
[2] 同志社大学理工学部
Faculty of Science and Engineering, Doshisha University
[3] 同志社大学大学院理工学研究科
Graduate School of Science and Engineering, Doshisha University
[a)] baba.takeru.bm1@is.naist.jp
[b)] emakihar@mail.doshisha.ac.jp

2. 関連研究

2.1 視線データの取得と解析

視線は，作業者が言語化できない無意識的な能力を反映し，視線の動き方が変化する．したがって，熟練者の持つ伝達不可能だったコツを，視線から明らかにできる．Busjahn らは，コードリーディングにおいて，注目した関数に関してマルコフモデルを生成した [2]．花房らは，プログラミング中の視線の分布と成績情報を基に，Support Vector Machine を用いて学習者のレベル分けを行なった [3]．

このように視線の振る舞いの違いから熟練者のコツを明

らかにし，初学者に教示することで作業効率が向上すると知られている．應治らは，ソースコードレビューにおける熟練度によるレビュー方法の違いを計測し，設計書やソースコードの読み方を作業者に教示した [4]．実験の結果，教示ありのグループは誤りの発見速度と発見率が向上した．

2.2 時系列データの連続動作分割

視線情報の時系列データから推移の特徴を調べる方法は，人が分割箇所を主観的に判断することが一般的である．もし時系列データの分割を数学的に行なうことができれば，大量のデータから連続的な行動の特徴を明らかにできる．時系列データの最も有名な分割手法に，隠れマルコフモデル（HMM）がある．HMM では直前の状態が次状態を決めるが，連続的な動作を 1 つひとつの状態へと離散化するため，分割箇所が細かく分かれ連続的な特徴を踏まえた分割ができない．長坂らは，形態素解析に用いられる n-gram の言語モデルを符号と動作に対応させた HDP-HMM+NPYLM により，連続的な動作の分節化をおこなった [5]．中村らは，ガウス過程と隠れセミマルコフモデルを組み合わせた，GP-HSMM の分割手法を提案した [1]．GP-HSMM は，離散化せずに各節をガウス過程で表現したことにより，HDP-HMM+NPYLM よりも高精度に動作の分節化ができると報告している．ただし，最大のクラス数を事前に指定する必要がある．

本稿では，実験結果の分析において，高速かつ高精度で分割できる GP-HSMM を用いた．この手法により分節化した視線の振る舞いが，熟練度ごとに異なる特徴を示すかを調査する．

3. 実験：デバッグ中の視線取得と解析

3.1 実験目的

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数学的な手法で分析し，熟練者の修正方針と初学者の躓きを調査することを目的とする．そこで，組込みシステムのデバッグ課題に関する被験者実験を行ない，デバッグ中の視線情報を記録，解析する．

3.2 実験タスク

実験タスクとして，スイッチの押下時に一定時間間隔で LED が点滅し，押下時以外は LED が消灯するシステムの構築を行なうデバッグ課題を設定した．被験者に与えるシステムには，ソースコードと回路にそれぞれ 2 つの誤りが初期状態で含まれている．ソースコード側には，動作の待ち時間に関する誤りと，回路の状態を正しく反映できるかを問う誤りがある．回路側には，接続確認を必要とする誤りと，電子部品の特性に関する知識が必要な誤りがある．

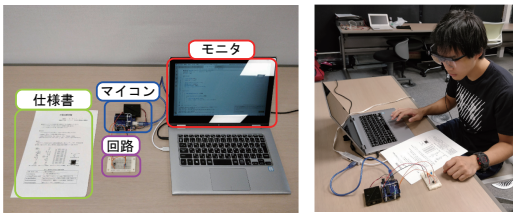

図 1　実験環境と実験の様子

3.3 実験環境

3.3.1 開発環境

実験は図 1 に示す環境で実施した．環境の構築に用いた組込みシステムの要素を下記に述べる．

- モニタ　：ソースコード編集や実行結果の確認に使用
- 仕様書　：回路図や実行方法，要件定義を示した用紙
- マイコン：回路を制御するマイコン（Arduino）
- 回路　　：デジタル入出力に関する回路

3.3.2 取得データ

被験者がシステムの誤りをデバッグをする間，ウェアラブルアイトラッカ（Tobii Pro Glasses 2）を使用して視線情報を 100 Hz で取得した．被験者視点映像のフレームレートは 25 fps で，解像度は 1920×1080 px である．視線が静止している fixation の状態と判定する条件は，60 ms の間視線の移動距離が 30 deg/s 以下だった場合である [6]．

3.4 実験手順

被験者実験は，下記に示す手順で実施した．

手順 1　被験者にシステム完成形を見せ課題の目標を説明
手順 2　被験者に課題内容やコンパイル方法を説明
手順 3　次の条件にあてはまる被験者には，仕様書の内容を見せながら，関数や回路図を簡潔に説明
　条件 1　Arduino 言語の使用経験なし
　条件 2　実体配線図や写真からのみ回路実装経験あり
手順 4　アイトラッカを装着しキャリブレーションを行う
手順 5　被験者はシステムのデバッグ課題を遂行
手順 6　組込みシステム開発歴や実験の感想をアンケート

3.5 被験者

組込みシステム開発経験がある 7 名の大学生に対して被験者実験を行なった．本研究では経験により獲得できる，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定義する．そこで，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．

3.6 実験結果

熟練者の修正方針や初学者の躓きを把握し，効率的なデバッグの手順を明らかにするために，数学的な分割手法である GP-HSMM を用いた [1]．GP-HSMM はランダム値

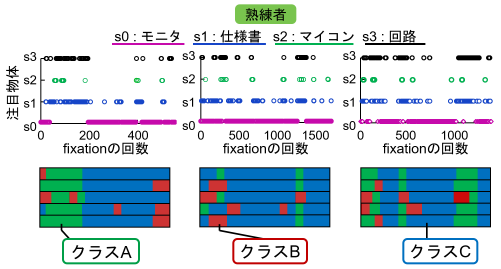

図 2　GP-HSMM による熟練者の実験時間の分割結果

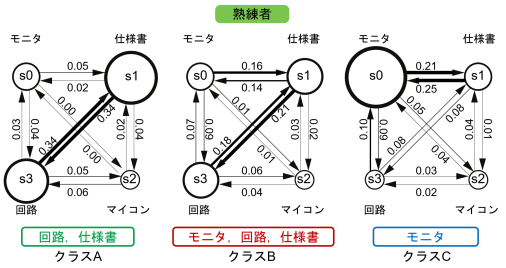

図 4　熟練者の 3 クラスにおけるマルコフモデル

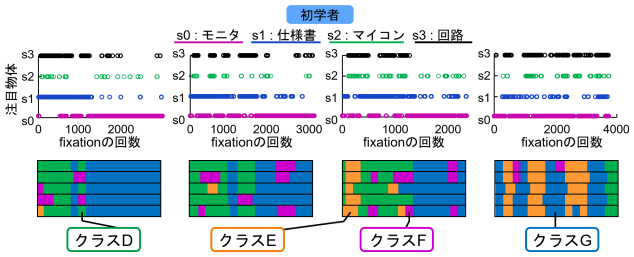

図 3　GP-HSMM による初学者の実験時間の分割結果

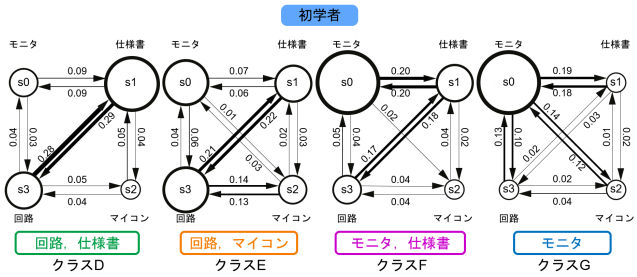

図 5　初学者の 4 クラスにおけるマルコフモデル

を用いて学習をするため，実行毎に結果が異なる．そこで，学習を 5 回試行し，それぞれの結果に共通する特徴を調査した．下記に時系列データ分割の手順を示す．

手順 1　4 種類の物体について注目時は 1，非注目時は 0 としてデータをサンプリング

手順 2　視線の移動距離を取得し各サンプルの値を 0 から 1 の範囲に正規化

手順 3　全点数を 100 サンプルにダウンサンプリング

手順 4　各被験者ごとに 100 行 5 列の行列に統合

手順 5　熟練度で 2 グループに分割し GP-HSMM を実行

手順 6　クラス番号はランダムにラベル付けされているため，目視によりラベルを正しく入換え

上記の手順で取得した時系列データから，実験開始から終了までにおける fixation ごとの注目物体の推移を調べた．注目物体は，我々の過去研究で提案した自動検出手法を用いて取得した [7]．注目物体として認識する物体は，3.3.1 項で示した 4 要素のみに制限する．s_0〜s_3 は，それぞれモニタ，仕様書，マイコン，回路のいずれかに被験者が注目している状態を示している．実験時間について 5 回ずつ分割を試行した際の，3 名の熟練者の分割結果を**図 2** に，4 名の初学者の分割結果を**図 3** に示す． 図上のグラフは横軸が fixation の回数で，縦軸がどの物体を見ていた状態かを示している．図下のグラフは，5 回の試行における分割箇所とクラスを色で示している．実験時間を分割した結果，熟練者は図 2 の 3 つのクラス，初学者は図 3 の 4 つのクラスに分割された．

各クラスの隠れ状態の意味は，人が判断しなければならない．作業内容判断のために時間分布のみに注目した場合，必然的に長くなる「モニタ」などに偏りが生じる．それに対し物体遷移は，注目時間と遷移回数に相関があるだけでなく，注目物体同士の関係性を表現できるため作業内容の判断に適している．以上の理由から，各クラスの注目物体の遷移確率をマルコフモデルとして出力することで，各クラスの行動を調べた．同じ物体へ注目が遷移する確率は非常に高く，また本研究では視線の遷移に着目しているため，異なる物体へ視線が遷移したデータのみを扱う．一般的にマルコフモデルの遷移確率は，ある状態からの出力確率の合計が 1 になるように算出する．しかし，遷移回数が少ない物体は高い確率を取り，不必要に目立つ問題がある．例えば，ある状態からの出力が 1 回だけの場合，遷移確率は 100%を示す．遷移回数が少ない状態は重要度が低いと考え，そのようなデータが目立たないように，モデル全体の遷移確率の合計が 1 となるように算出した．ある状態 s_a から s_b への遷移 p_{ab} について $(a, b \in \{0,1,2,3\}, a \neq b)$ とする．また，計測時間内の遷移回数を n_{ab} とする．この表記を用いると，全体の遷移回数は次の式で示される．

$$n_{all} = \sum_{x \in X} \sum_{y \in Y} n_{xy} \ (X, Y = \{0,1,2,3\},\ x \neq y) \qquad (1)$$

クラス a から b への遷移確率 p_{ab} は次の式で示される．

$$p_{ab} = \sum_{x \in X} \sum_{y \in Y} \frac{n_{ab}}{n_{xy}} = \frac{n_{ab}}{n_{all}} \qquad (2)$$

熟練者の 3 クラスに関するモデルを**図 4** に，初学者の 4 クラスに関するマルコフモデルを**図 5** に示す． 図の矢印の太さは，遷移確率の高さを表したものであり，矢印が太いほど確率が高い．また，注目時間が長いほど，注目状態を示す円が大きくなる．モデリングの結果から，熟練者・初学者の各クラスにおける注視物体を調査した．その結果，熟練者は図 4 のクラス A が「回路と仕様書」，クラス B が「モニタと仕様書と回路」，クラス C が「モニタ」にそ

れぞれ注目する状態だった．初学者は図 5 のクラス D が「回路と仕様書」，クラス E が「回路とマイコン」，クラス F が「モニタと仕様書」，クラス G が「モニタ」にそれぞれ注目する状態であることが分かった．

4. 考察

図 2 や図 3 で示した GP-HSMM による実験時間の分割結果から，熟練者は 3 つのクラス，初学者は 4 つのクラスでデバッグを進めることがわかった．熟練者の各クラスの行動を図 4 の仕様書から出る遷移確率から推定すると，仕様書を参考に回路を修正するクラス A の行動と，仕様書を参考に回路とソースコードを修正するクラス B の行動，ソースコードを修正するクラス C の行動が存在すると考えられる．熟練者の 3 種類の行動と図 2 を同時に考慮すると，熟練者は序盤に回路，中盤にソースコード，終盤に両方を見てシステムの動作確認をする方針があるとわかった．これは実験後のヒアリングにおいても，同様の修正方針の回答が得られた．

初学者の各クラスの行動を図 5 の仕様書から出る遷移確率から推定すると，仕様書を参考にソースコードを修正するクラス D の行動と，マイコンと回路を見比べながら修正するクラス E の行動，仕様書を中心にソースコードを修正するクラス F の行動，ソースコード修正するクラス G の行動が考えられる．4 種類の行動を修正箇所でまとめると，クラス D と E が回路に関する修正，クラス F と G がソースコードに関する行動である．4 種類の行動と図 3 を同時に考慮すると，初学者は前半に回路，後半にソースコードを修正する傾向があった．しかし，序盤において回路とソースコード間で注目を繰り返しており，なおかつデバッグ方針に関するヒアリングでは「どちらからシステムを修正するか迷った」などの回答が得られ，明確な修正方針に従う行動ではなかった．また，システムの動作確認時の方針に関するヒアリングでは「時と場合によって確認箇所が変わる」といった回答が得られた．したがって初学者は，システムの確認方法を知らない可能性が考えられる．以上の結果から，GP-HSMM の分割は初学者の実質的な躓きは明らかにできなかったが，潜在している熟練者の修正方針を顕在化することができた．

この熟練者の結果は，過去の論文における目視による行動内容の推測と同じ結果を示している [8]．明らかになった熟練者の 3 段階は，過去の論文で示した「回路が修正できていなければプログラムを試せない」という序盤で回路をデバッグする方針と，システムの動作確認時に「HW と SW を同時に確認しながら修正する」という終盤における方針を包括した 3 段階であると考えられる．

初学者の結果と，過去の結果を同時に考えると，初学者の躓きを 2 種類に分類できる [8]．1 つ目の躓きは，デバッグの修正方針を持たないことである．序盤において修正箇所を回路とソースコード間で遷移を繰り返し，なおかつ回路が修正されていない状態でソースコードのデバッグをしており，熟練者のように決まった修正方針を持たない．2 つ目の躓きは，システム全体の動作確認方法を知らないことである．組込みシステムは，1 箇所の HW のバグが SW の動作を全て隠蔽することがあり，動作確認を設けて熟練者のように双方の要素を同時に考慮するべきである．

5. 結論

本研究では，組込みシステム開発のデバッグにおける熟練度ごとの特徴を，視線動作から調査した．実験の結果，各熟練度におけるデバッグの修正方針を明らかにした．GP-HSMM により分割とクラスタリングをした結果，デバッグの段階は熟練度によって異なった．熟練者は回路，ソースコード，システムの動作確認の 3 段階でデバッグを行なう．初学者は回路，ソースコードの 2 段階でデバッグを行なう傾向にあったが，GP-HSMM の分割結果だけでは躓きを明らかにすることができなかった．しかし，ヒアリング結果などを考慮すると，初学者は明確な修正方針や確認方法を知らないといった躓きが明らかになった．

また，修正方針/動作確認方法を知らない初学者に対するアドバイスを，本研究で得られた熟練者の修正方針やデバッグのコツをもとに提案できると考えられる．熟練者の特徴を初学者に教示することで実際にデバッグ効率が向上するかを調査することが，本研究の展望である．

参考文献

[1] Nakamura, T., Nagai, T., Mochihashi, D., Kobayashi, I., Asoh, H., and Kaneko, M.: Segmenting continuous motions with hidden semi-Markov models and Gaussian processes. In Frontiers in Neurorobotics, 2017, pp. 67–83.
[2] Busjahn, T., Schulte, C., Sharif, B., Simon, B., Begel, A., Hansen, M., Bednarik, R., Orlov, P., Ihantola, P., Alperovich, G., and Jetbrains, M.: Eye Tracking in Computing Education. In Proceedings of the 10th Annual Conference on International Computing Education Research, 2014, pp. 3–10.
[3] 花房 亮, 山岸秀一, 松本慎平, 加島智子：機械学習処理に基づいたプログラミング読解中の視線軌道の自動分類. 人工知能学会全国大会論文集, 2015, pp. 1–2.
[4] 應治 沙織, 上野 秀剛：コードレビュー時の読み方教示によるレビュー効率の変化. 研究報告ソフトウェア工学（SE）, 2014，pp. 1–8.
[5] 長坂 翔吾, 谷口 忠大：階層 Pitman-Yor 言語モデルを用いた動作解析. 人工知能学会全国大会論文集, 2011, pp. 1–4.
[6] Olsen, A.: The Tobii I-VT fixation filter. Organized by Tobii Technology AB, 2012.
[7] 米田 浩崇, 槇原 絵里奈, 馬場 建, 前田 侑哉, 三木 光範：組込みシステム開発における YOLO v3 を用いた注目物体検出手法の提案. 同志社大学ハリス理化学研究報告, 2020, pp. 36–40.
[8] 馬場 建, 槇原 絵里奈, 米田 浩崇：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. 研究報告ソフトウェア工学（SE）, 2019, pp. 1–8.