社団法人 電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.
SS96-62 (1997-03)

ソフトウェア開発者の運動情報に基づく特徴的行動の検索方法の提案

大澤 文男, 島 和之, 松本 健一, 鳥居 宏次

奈良先端科学技術大学院大学 情報科学研究科

〒630-01 奈良県 生駒市 高山町 8916 番地の 5

matumoto@is.aist-nara.ac.jp

あらまし ソフトウェアの品質や生産性に影響する要因を明らかにする研究が多くなされている．その中で，ソフトウェア開発者の行動をビデオで記録し，分析する研究がなされている．しかし，開発者の特徴的行動を検索するためにはビデオデータを繰り返し見る必要があるので、分析に多大な時間がかかるという問題がある．

本研究では、運動情報を用いることにより開発者の特徴的な行動を検索する手法を提案する．提案する手法では，フーリエ記述子を用いて与えられた動きに近い開発者の動きが現れた時刻が求まる．実験の結果、提案する手法を用いることにより，分析に要する時間を短縮できることが示された．

キーワード ソフトウェア工学, フーリエ記述子, 特徴抽出, プログラミング行動分析

A Method for Searching Characteristic Behavior Based on Software Developers' Movement

Fumio OSAWA, Kazuyuki SHIMA, Ken-ichi MATSUMOTO, and Koji TORII

Graduate School of Information Science, Nara Institute of Science and Technology

8916-5 Takayama-Cho, Ikoma-Shi, Nara 630-01
matumoto@is.aist-nara.ac.jp

Abstract

Much research takes place concerning factors that affect software quality and productivity. One such research concerns the analysis of video data of software developers' activity. However, it takes a lot of time since it requires repeated viewing of the video data for searching characteristic behavior of developers.

This paper proposes a method for searching the characteristic behavior by the description of movement. It finds the time in which the developers' behavior showed movement similar to movement described with the Fourier descriptor. An experiment showed that using our method will result in shortening the time necessary to analyze video data.

key words software engineering,Fourier descriptors,extraction of characteristic s,programing behavior analysis

1 はじめに

ソフトウェアの品質や生産性に影響する要因やその影響の程度を明らかにするために，ソフトウェア開発者の行動を観測，分析する研究が行われており，開発者の行動を記録したビデオやキーストロークデータ（開発者が入力したキーの時系列データ）などが収集されている．

例えば，Mayrhauser ら [10] は，プログラムを理解する行動の細分化と保守担当者が必要とする情報の抽出を行った．また柳ら [7] は，キーストロークデータを記録・分析することによって，誤りが混入するときのデバッグ作業者の特徴的な行動を明らかにした．また，高田ら [6] は，キーストロークデータを記録・分析することによって，誤りの発見と修正の行動を細分化した．三輪ら [8] は，エディタ操作時のキーストロークデータを記録・分析することによって，操作の習熟の様子を明らかにした．津田ら [9] は，ソフトウェア変更作業における，作業工程を分析することによって，作業に必要な情報や作業内容を明確にした．

キーストロークデータを分析することにより，開発者が計算機を用いてどのような作業を行っていたかを調べることができる．例えば，各時刻に入力されたキーストロークを調べることにより，開発者が，プログラムのコンパイルを行っている時刻や，プログラムの修正を行っている時刻などを特定することができる．このようなキー入力を伴う開発者の行動の特定には，ツールを用いてキーストロークを自動的に分析することが可能である [6][7]．

しかし，開発者の行動はキー入力を伴うものばかりではない．例えば，開発者は “本を読んでいる” 時や “考え込んでいる” 時，開発者はキー入力を行わないので，キーストロークデータのみからでは，これらの 2 つの行動を区別することはできない．開発者がキー入力を行っていない時に，どのような作業を行っているのかを知るためには，作業中の開発者をビデオカメラで撮影し，実験後に分析者がその映像を見て判断する必要がある．これらの場面は，ビデオデータを分析することにより特定することが可能である．しかし，そのためにはビデオデータを繰り返し見る必要があり，実験時間よりも長い時間がかかるという問題が指摘されている [13][12]．

そのため，キーストロークデータやビデオデータなどを収集し，効率的に分析するためのシステムとして Ginger2 System が開発されている．Ginger2 System は，開発実験において収集された様々なデータを視覚化し，同期して再生したり，分析に必要な場面を検索したりするためのシステムである．

本研究では，キーストロークデータのみからでは特定できない行動を，分析者がビデオデータを調べて特定する場合に，分析者が注目している行動を効率的に検索するための方法を提案する．本研究で提案する手法を用いることにより，分析者が例示した行動と同様の行動を開発者が行っている場面を検索することができるようになる．

提案する手法では，作業中の開発者の体の各部の動きを計測し，その特徴を抽出することにより，特徴的な動作が現れた時刻を特定する．動きの特徴を抽出するための方法としてはフーリエ記述子を用いる．フーリエ記述子は，文字，飛行機，機械部品などの形状を識別するための方法として用いられている．また，フーリエ記述子を用いて，筆跡の動的情報より筆跡の特徴を抽出し，筆者を識別する手法の研究がなされている [1]．

筆者識別のための手法では，予め登録されている筆跡と筆者の筆跡との近似の度合を求めるためにフーリエ記述子を用いる．本研究で提案する手法では，分析者によって例示された開発者の動作と作業中の開発者の動作との近似の度合を求めるためにフーリエ記述子を用いる．

実験の結果，本方法によって開発者の特徴的な動きが現れた時刻を，自動的に特定することができ，分析に要する時間を短縮できることを示した．と言える．

以降，2 章では，Ginger2 System と本研究の位置付けについて述べる．3 章では，動きの特徴抽出方法について述べる．4 章では，提案手法の有効性の評価のために行なった実験とその結果について述べる．5 章は，まとめと今後の課題である．

2 Ginger2 と研究の位置付けについて

本章では，Ginger2 System[11] の概要と本研究の位置付けについて述べる．

Ginger2 System は，被験者が計算機を使用して行なう作業に対して，作業の観測に用いるデータを収集・分析するためのシステムである．Ginger2 System の特徴の一つとして，複数の機材を用いてデータを同時に収集することができる．ここでは，収集可能なデータの種類と機器について述べる．

2.1 収集可能なデータ

Ginger2 System で収集が可能なデータの種類には次のものがある．

- 計算機の画面の変化
  - ウィンドウの開閉，位置，大きさの変化
  - 各ウィンドウ内のテキストカーソルの移動
  - 各ウィンドウ内のテキストの変化（フォントや色や装飾の変化は除く）
  - 各ウィンドウ内の graphic object の変化
  - ウィンドウの重なりの変化
- 計算機に対する被験者の操作
  - マウスの移動

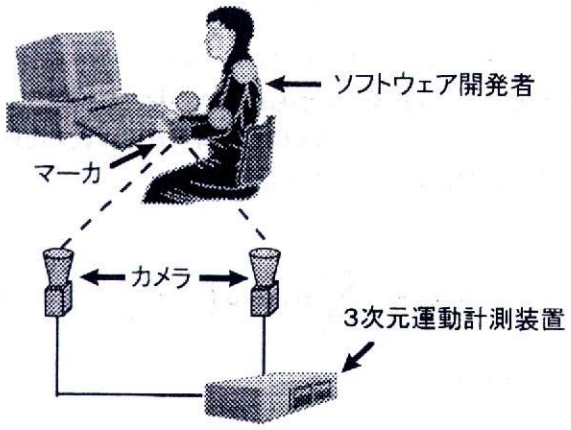

図 1: 3次元運動計測装置

- マウスボタンの操作
- キーストローク

- 実験者のメモ
  - 実験中の被験者が気付いたことをその場で残したメモ
- 映像と音声
  - 被験者の挙動や表情
  - 発話
- その他
  - 被験者の注目している計算機画面上の点（注視点）の遷移
  - 被験者の手や頭や肩などの動き

2.2 3次元運動計測装置

本研究では，Ginger2 System で収集されるデータの中から，3次元運動計測装置から得られる開発者の動きの情報を用いている．ここでは，3次元運動計測装置について述べる．

開発者の動きの計測には，応用計測研究所 (OKK INC.) の3次元運動計測装置 (Quick MAG SystemIII)[14] を用いている．図1は，3次元運動計測装置による，計測の様子を示した図である．3次元運動計測装置は，マーカーを2台のカメラで計測し，それぞれのカメラで得られた画面上の座標から実空間座標を得るものである．マーカーを開発者の体につけ，マーカーの位置座標の変化を計測することにより，開発者の体の動きを捉えることができる．

3次元運動計測装置は，それぞれのカメラの画面内でマーカーの色に近い色をもつ領域を識別し，その中心座標をマーカーの画面上の座標とする．個々のマーカーの色は任意の色に設定することができるが，他のマーカーの色や背景にある物体の色と近い色に設定すると，それらを混同してしまい，マーカーの位置を正しく計測できなくなる．本研究では，マーカーの色として互いに区別しやすい，青，赤，緑，黄，ピンク，オレンジを用いた．また，被験者の服の色とマーカーの色とを混同することを避けるために，被験者に白または黒の上着を着てもらった．

本研究では，ソフトウェア開発者が椅子に座り，計算機のディスプレイとキーボードとマウスを乗せた机の上で作業を行う場合を想定して，開発者の特徴的行動の抽出を試みた．この場合，開発者は椅子に座っているため，胴や下半身はあまり動かないと考えられる．そこで，ソフトウェア開発者の腕の動きに着目して計測を行った．

腕の動きを計測するためには，体の前方にカメラを配置する必要がある．しかし，正面に配置すると計算機のディスプレイが邪魔になるので，開発者の右斜め前と左斜め前にカメラを配置した．3次元空間における物体の位置と向きを決定するためには，一直線上にない3箇所を計測する必要がある．本研究では，左右のそれぞれの腕について，肘，手首上面，手首側面の3箇所を計測した．手首側面の片側は片方のカメラからしか計測できないので，側面の左側と右側には同じ色を用いた．つまり，右斜め前のカメラでは左右手首の右側面のマーカーを計測し，左斜め前のカメラでは左右手首の左側面のマーカーを計測した．このようにすることにより，右のカメラから右側面のマーカーへの直線と，左のカメラから左側面のマーカーへの直線の交点の座標を得ることができる．この交点は，手首のほぼ中心に位置していると考えられる．

2.3 実験環境

Ginger2 System による実験の環境を，図 2に示す．キーストロークデータと注視点を記録した行動履歴データ，開発者の表情・挙動・発話内容を記録した映像データ，そして開発者の挙動の数値データである運動データを自動的に収集し，分析の際には，複数のデータを同期再生することができるので，それらを組み合わせることにより，単独のデータによる分析よりも詳細な分析が可能となり，単独のデータではわからない部分の情報を補いあったり，データ間の関係を調べることができる．

2.4 検索方式

本研究は，Ginger2 System の検索方式を設計するために，開発者の動きに関するデータを用いた特徴抽出・検索手法を提案し，その手法による実験を行なった．

ビデオデータの分析を行なううえで，見たい場面を効率良く検索できることが望まれる．Ginger2 System は，複数のデータを扱っているため，複数のデータにまたがった検索条件を入力したり，ある種類のデータから別の種類のデータを検索するなど，さまざまな検索の入力方法の要求があると考えられる．

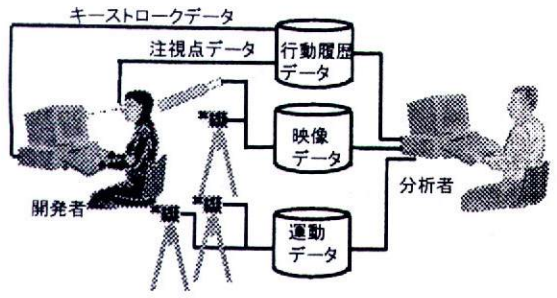

図 2: 実験環境

図 3は，検索方法の例である．ある行動記録の中から，検索したい動き X の範囲を指定してやると，動き X と同じ動きの時間が検索結果として得られる．特徴抽出の手法は，後述するフーリエ記述子を適用する．

本研究で提案する手法は，開発者の運動情報に用いて，与えられた動きに近い開発者の動きを検索する．この手法を用いることにより，まったく同じ動きだけでなく，似た動きも捜し出すことが可能である．

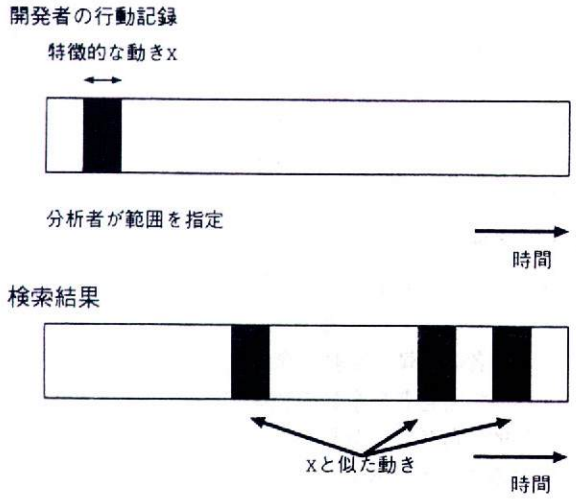

図 3: 検索例

3 動きの特徴抽出

本章では，開発者の動きどうしを区別するために，動きの特徴を反映したベクトル（以下パターンベクトル）を抽出する方法について述べる．このパターンベクトルを用いて，開発中に計測された複数の動きの中から，分析者が探したい，開発者の特徴的な動きを検索する．

3.1 フーリエ記述子

平面上の曲線を周波数領域で記述する方法としてフーリエ記述子がある．このフーリエ記述子は，文字，飛行機，機械部品などの形状を識別する方法として用いられている．

それは，曲線を作る点列そのもののフーリエ変換を考えたもので，曲線を一点からの長さの複素数値関数と見なして，これをフーリエ展開して得られる係数を記述子とする．曲線 C のフーリエ記述子は，

$$e(k) = \frac{1}{n}\sum_{j=0}^{n-1} z(j)\exp\left(-2\pi i\frac{jk}{n}\right) \qquad (1)$$

と表される．

3.2 パターンベクトルの生成

サンプル時刻毎に抽出された開発者につけた各部所のマーカーの位置座標を x-y 平面，y-z 平面，x-z 平面に分解し，それぞれの平面について，マーカーの位置座標からフーリエ記述子を求める．このとき，本を読むといった被験者の動きをある単位動作毎に区切り，フーリエ展開を行なう．つまり，複数の単位動作の組合せで，一つの特徴的な動きを構成しているとする．この単位動作は，ここでは，マーカーをつけた各部所の移動のベクトルの向きが逆方向になった時から，次にベクトルの向きが逆方向になるまでとする．そして，得られた記述子の n 次以下の実数成分と虚数成分を成分として持つベクトルをパターンベクトルとする．すなわち，N 次の複素フーリエ係数の実数部分を a_i，虚数部分を b_i として，$(a_i, b_i), i = 0, 1, \cdots, N/2-1$ を複素フーリエ成分の低次項，$(a_i, b_i), i = -1, -2, \cdots, -N/2$ を複素フーリエ成分の高次項とすると，$n(< N/2)$ 次以下のフーリエ係数により生成されるパターンベクトル X は，$X = (a_0, b_0, a_1, b_1, a_{-1}, b_{-1}, \cdots, a_n, b_n, a_{-n}, b_{-n})$ の (4n+2) 次元ベクトルで表せる．但し，パターンベクトル X の各成分について，それらの絶対値換算での最大値で正規化している．また，記述子の低域成分に原曲線の大まかな形の情報が含まれているので，低域成分のみを用いる．このようにして，動きの特徴を持つパターンベクトルを生成する．

3.3 クラスタリング

次に，パターンベクトルを区別するための基準となる，参照ベクトルについて述べる．参照ベクトルは，多数のパターンベクトルを，各単位動作のパターンベクトルの分布状態に応じて少数のベクトルに置換したものである．それぞれのパターンベクトルが，動きの特徴を反映したものならば，同じ動きのパターンベクトル同士はパターン空間上で近接すると考えられる．そのため，パターンベクトルの分布密度が高いところを参照ベクトルで置換することにより，参照ベクトルを各動きを区別す

るための基準とする．パターンベクトルが得られた時，最も近い位置にある参照ベクトルが表す動きが，このパターンベクトルの動きであるとする．

各動きを適切に区別するためには，パターンベクトルの数や分布に応じて適切に参照ベクトルを配置する必要がある．そこで，k-mean クラスタリングアルゴリズムを用いて，参照ベクトルを配置する．

図4は，パターン空間における，パターンベクトルとクラスタリングされた参照ベクトルを示した図である．同一パターンベクトルの分布の中心に，参照ベクトルが配置されている．

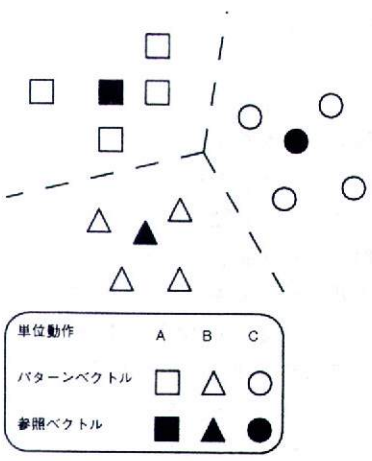

図 4: パターンベクトルと参照ベクトル

3.4 動きの識別

計測された被験者の動きの情報から得られたパターンベクトルから，動きを識別するには，まず，得られた多数のパターンベクトルを，あらかじめ決定しておいた少数の参照ベクトルを用いて分類する．ある動きからパターンベクトルを求めた時，そのパターンベクトルとの距離が最も近い参照ベクトルに対応した動きが，そのときの被験者の動きの特徴を表した動きとなる．検索対象が参照ベクトルという，動きを代表する特徴量であるため，全く同じ動きだけではなく，近い動きも検索することができる．

これらのパターンベクトルや参照ベクトルは，ある期間（ここでは，開発者につけられた各マーカーの移動方向が逆向きになった時から，次に，逆向きに変わるまで）における開発者の動きを反映しており，これらの組合せで，本を読んでいる等の特徴的な動きが構成される．そこで，例えば「本を読んでいる場面」の検索は，その動きに対する参照ベクトルの状態の遷移を調べ，そこで得られた状態遷移を計測データから検索することにより可能となる．

4 評価実験

この章では，ソフトウェア開発者の特徴的な動きの検索に対する，提案手法の有効性を評価するために行なった実験について述べる．今回，特徴的な動きの中から，開発の初心者が困っている場面について着目し，評価実験を行なった．

4.1 予備実験

開発者が困っている場面を識別するためには，対象となる動きを設定する必要がある．ここでは，その動きを決めるために行なった実験について述べる．

4.1.1 実験手順

まず，識別すべき開発者の動きを決定するために，次の予備実験を行なった．行なった実験は，C 言語の初心者に課題を与え，それに基づいてプログラムしてもらうというものである．実験を行なう際，被験者には，視線の方向を計測するためにアイカメラを，そして3次元計測装置で動きを計測するために，被験者の両肘，両手首，両手首側面の計8箇所にマーカーを付ける．そして，実験中の被験者の開発の様子と計算機の画面をビデオカメラで記録する．実験中は，被験者が今何をしようとしているのか，何に対して悩んでいるのかなど考えていることを絶えず発話してもう．

実験で用いた課題の内容は，文字列操作に関するもので，課題1は「標準入力から入力された文字列を標準出力で出力する.」，課題2は「標準入力から入力された文字列を，スペースまたはタブを区切りとして，分ける.」というものである．被験者には，仕様書，筆記用具，C 言語の参考書を与える．仕様書は左側に，参考書は右側に置き，必要に応じて使用してもらう．開発中の動きを取ることが目的であったため，関数の使い方がわからないなど問題点に直面した時には，man コマンドは使用せずに，参考書で調べてもらう．開発の進行が著しく滞り，参考書を見ても直面した問題を解決できそうにない場合は，観察者がアドバイスを与える．

また，この実験では，参考書や仕様書を読んだ後，もとの場所に戻すという制限を設ける．これは，手元に参考書や仕様書を置いてキー入力を続け，参考書等が必要な時に視線だけで手元の参考書を読んだ場合，ビデオデータを見ている分析者には，動作の区切りがわかりにくく，パターンベクトル抽出が正しく行なわれているかの判断がつきにくいからである．

4.1.2 実験結果

C 言語初心者の学生二人に被験者になってもらい，実験を行なった．被験者1に対して課題1と課題2を，そして被験者2に対して課題1を与えた．

表1は，それぞれの被験者が特徴的な動きを行なった

	被験者1	被験者2
本を読む	3／3	6／6
キーを入力する	0／39	0／28
マウスを操作する	0／27	0／16
顎をさわる	0／0	3／3
画面を指さす	0／1	0／0

表 1: 困っている状態と各動作の関係

時と困った状態との関係を表している．分母が，その動作を行なった総数で，分子が困っている状態の時に行なった回数である．これより，困っている可能性が高いのは，本を読む動作のときということがわかる．

そこで，困っている動作として，本を読むという動作を用いることにした．また，被験者が行なった他の動作として，ウィンドウを切替えるためにマウスに手を伸ばす，キーを叩いている，顎をさわる，画面を指さす，画面をじっとみる動作が観測された．次の節では，提案手法により，本を読む動作をこれら他の動作から識別できるかを評価するために行なった実験について述べる．

4.2 評価実験

ここでは，提案手法が，開発者が開発中にとる動作の中から本を読む動作の識別の有効性を評価するために行なった実験について述べる．

4.2.1 目的

提案手法の有効性を評価することが，この実験の目的である．ここでは，ある動きを検索する場合に，その動きをどれだけ見落さずに探し出すことができるかを，有効性の評価の基準とする．

分析者が，分析を行なうために，ある動きを検索するとする．そして，その検索結果をもとに場面を再生するとき，誤って検索した場合はその検索結果を除外すればよいが，見落しの場合は，その動きがあったことを分析者は知ることができない．そのため，誤認識よりも，見落しの割合の方が重要だと考える．

そこで，どの程度見落しがないかを評価基準とする．

また，検索にかかる時間も評価基準となる．ビデオデータによる検索よりも時間がかかっていたのでは，最初からビデオデータを用いれば良いのだから，提案手法の意味がなくなってしまう．提案手法を用いることにより，検索時間がどの程度短縮することができるのかも評価基準として考慮する必要がある．

4.2.2 実験手順

被験者に開発中にとる動作を繰り返し行なってもらい，提案手法により，それらの動作を識別できるかを評価する実験を行なった．

被験者には，3次元計測装置で動きを計測するために，マーカーを被験者の両肘，両手首，両手首側面の計8箇所につけた．そして，実験中の被験者の開発の様子をビデオカメラで記録した．

被験者が行なう行動は，予備実験で観測された，本を読む，画面を指さす，キーボードでキーを打っている，顎を触る，マウスを使うという5種類の動きである．

また，実験では，動作を2つ一組みとした．これは，ある時点での動きは，前後の動きに影響を与える可能性があるが，それより前の動きや，後の動きとは影響を与え合うことはないと思われるからである．動きの組合せは次のとおりである．

- 画面を指さす → 本を読む
- 本を読む → 顎を触る
- 顎を触る → 本を読む
- 本を読む → 画面を指さす
- 画面を指さす → 顎を触る
- 顎を触る → 画面を指さす
- 画面を指さす → キーを入力する
- キーを入力する → マウスを操作する
- マウスを操作する → キーを入力する
- キーを入力する → 本を読む
- 本を読む → キーを入力する
- キーを入力する → 顎を触る
- 顎を触る → キーを入力する
- キーを入力する → 画面を指さす
- 画面を指さす → マウスを操作する
- マウスを操作する → 本を読む
- 本を読む → マウスを操作する
- マウスを操作する → 顎を触る
- 顎を触る → マウスを操作する
- マウスを操作する → 画面を指さす

実験は，各動作を10秒ごとに連続的に行った．つまり，「画面を指さす」→「本を読む」→「顎を触る」→「本を読む」‥という具合に，進めていき，一番下の画面を指さすまでいったら，一番上に戻り，「本を読む」→「顎を触る」と続ける．実験は，一番上から一番下までを3回繰り返し，各動作4回×3回の計12回行なった．

そして，得られた運動データからパターンベクトルを抽出し，それをもとにして動きの識別を行なう．

4.2.3 実験結果

図5は，「本を読む」動きの参照ベクトルと各パターンベクトルとの距離を示している．横軸は時間を示しており，数字は，「本を読む」動作を行なった回数である．縦軸は，「本を読む動作」からの距離である．距離が近いほど，「本を読む」動きに近い動きであることを示して

いる．

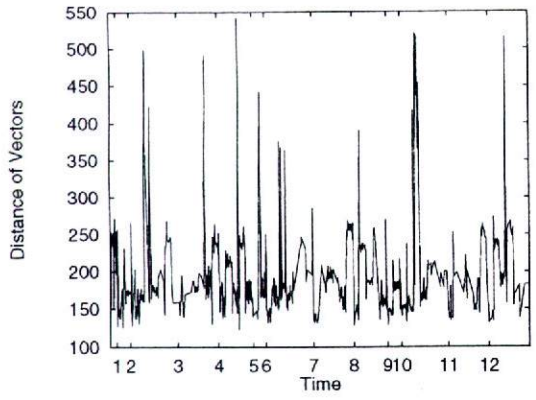

図 5: 参照ベクトルと各パターンベクトルとの関係 1

図 6の横軸は，図 5と同様に時間を示している．そして縦軸は，各パターンベクトルについて参照ベクトルからの距離を考え，近い順に並べた時の順位の逆数を表している．つまり，値が高いほど順位が小さいので，より「本を読む」に近い動きであると考えられる．

60 動作中「本を読む」動作は 12 回行なわれている．図 6について，全ての本を読む動作を含むように閾値を設定したとすると,「本を読む」と識別するのは，22 回（そのうち誤識別 10 回）である．つまり，45%の誤検出を行なっている．

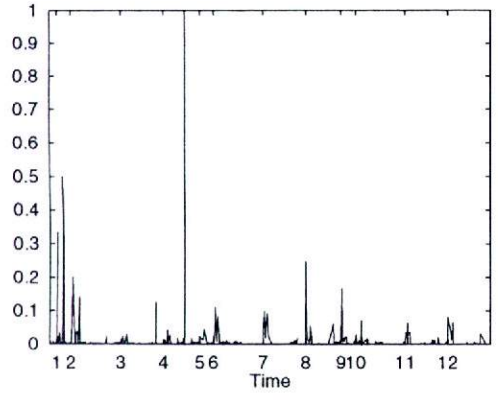

図 6: 参照ベクトルとパターンベクトルとの関係 2

予備実験において，被験者 1 の場合，59 分 4 秒の間に 3 回本を読んでおり，被験者 2 の場合は 25 分 49 秒の間に 6 回本を読んでいる.「本を読む」動作は，1 時間 24 分 53 秒の間に合計 9 回含まれている．

予備実験においても，45%の誤検出を行なうと考えると，17 場面分析する必要がある．

ビデオデータで「本を読んでいる」場面を探す場合，探したい場面の場所がわからないためビデオデータを最初から最後まで見る必要があり，約 1 時間半もの時間がかかってしまう．

本手法を用いることにより，探したい場面の時間を特定が可能である．一場面検索するのに 30 秒かかる．誤認識した場面も含めて 17 場面全て検索しても，8 分 30 秒で検索可能なので分析に要する時間を短縮できると予想される．

5 まとめと今後の課題

本稿では，ソフトウェア開発者の開発動作を識別するために，フーリエ記述子を適用した動きの特徴抽出手法を用いて，開発動作から動作を検索する手法について述べた．

そして，評価実験を行ない，本手法を用いることによって，探したい場面の検索時間を短縮できることを確認した．

場面がわかることにより，教育などに用いることが可能である．

今回提案した検索手法により，動きの区別が可能となったので，Ginger2 System で収集可能な他のデータとの組合せも考慮することにより，今後，より効果的な分析手法，分析ツールが期待できる．

参考文献

[1] 山崎 恭, 小松 尚久: “カテゴリー化された筆跡情報に基づく個人性抽出手法, ” 信学論, **J79-D-II**, 8, pp. 1335 – 1346(1996).

[2] 上坂 吉則: “開曲線にも適用できる新しいフーリエ記述子,” 信学論, **J67-A**, no.3, pp.166 – 173,(Mar. 1984).

[3] 原島 博監修: “画像情報圧縮,” 第 6 章, pp.115 – 139, オーム社 (1991).

[4] T.Kohonen: “Self-Organization and Associative Memory (second edition),” Springer-Verlag (1988).

[5] 鳥脇 純一郎: “認識工学-パターン認識とその応用,” pp.62 – 108, コロナ社 (1993).

[6] 高田 義弘, 鳥居 宏次: “プログラマのデバッグ能力をキーストロークから測定する方法,” 信学論, **J77-D-I**, 9, pp. 646 – 655 (1994).

[7] 柳 正純, 高田 義弘, 鳥居 宏次: “キーストロークに着目したバグ混入時のプログラミング行動の特徴の分析,” 第 49 回情処全大, pp. 5-511–5-512 (1994).

[8] 三輪 和久, 櫻井 桂一, 岡田 稔, 岩田 晃: “初心者のプログラミング行動時系列分析,” 信学技報, ET93-47, pp. 9–16 (1993).

[9] 津田 道男, 永岡 郁代, 青木 一紀, 和栗 正一: “ソフトウェア変更作業の分析と支援機能,” 情処学ソフトウェア工学研報, Vol. 95-SE-11, pp. 1–6 (1995).

[10] A.V. Mayrhauser and A.M. Vans: “Comprehension processes during large scale maintenance,” Proc. 16th Int. Conf. Soft. Eng., pp.39-48 (May 1994).

[11] 松本 健一, 島 和之, 高田 義弘, 高田 眞吾: “Ginger2 Home Page,” http://hawk.aist-nara.ac.jp/ginger2/index-e.html.

[12] 門田 暁人, 高田 義広, 鳥居 宏次: “視線追跡装置を用いたデバッグプロセスの観察実験,” 信学技報, Vol. SS96-5, pp. 1–8 (1996).

[13] 小林 淳, 下條 善史, 松本 健一, 鳥居 宏次: “ソフトウェアの開発/利用行動を記録した映像に対する自動索引付け法の提案,” 情処学会研報, 96-SE-107,pp. 33-40 (1996).

[14] (株) 応用計測研究所: “Quick MAG SystemIII ユーザーズマニュアル”