様　式　Ｃ－１９、Ｆ－１９、Ｚ－１９　（共通）

科学研究費助成事業　　研究成果報告書

平成　２８　年　　６　月　　８　日現在

機関番号：　１４４０１
研究種目：　若手研究(A)
研究期間：　2012～2015
課題番号：　２４６８０００２

研究課題名（和文）超大規模ソースコードを対象としたコードクローン検出システムの構築

研究課題名（英文）Method and Implementation of Code Clone Detection for a Huge Set of Program Source Code

研究代表者

肥後　芳樹（Higo, Yoshiki）

大阪大学・情報科学研究科・准教授

研究者番号：７０４５２４１４

交付決定額（研究期間全体）：（直接経費）　　7,500,000円

研究成果の概要（和文）：多数のソフトウェアの集合に対して高速にコードクローンを検出する手法を提案した．約3億行のソースコードから2時間程度でメソッドレベルのコードクローンを検出することができた．従来はファイル単位のクローンしか検出できておらず，提案手法を用いることにより従来は検出されることがなかった多数のクローンが検出できることがわかった．また，開発履歴データを解析することにより高速にコードクローンの変遷を追跡する手法を考案した．5,000リビジョンの開発履歴データから3時間程度で追跡が行えることを確認した．また従来手法では追跡できなかった多数のコードクローンが追跡できていることも確認した．

研究成果の概要（英文）：We have proposed a technique to detect code clones (hereafter, clones) from a large set of software projects. Our technique detects method-level clones while existing techniques detect file-level clones. We confirmed that our technique can finish detecting clones from 300 million lines of code. We also have proposed a technique to trace clones along development history of source code. Our technique can trace clones even if its location in a system was changed while existing techniques cannot. We confirmed that our technique takes about 3 hours to trace clones along 5,000 revisions of source code history.

研究分野：　ソフトウェア工学

キーワード：　コードクローン　ソースコード分析

１．研究開始当初の背景

ソースコード中のコードクローンの存在は，古くからソフトウェアの保守性を悪化させる要因の 1 つであると考えられてきた．例えば，あるコード片からバグが検出された場合を考える．もし，そのコード片のコードクローンがシステム中に存在した場合には，同様のバグがそのコードクローンにも存在している可能性がある．もし，保守監理者がそのコードクローンの存在を知らない場合は，バグを取り除いたつもりであっても，システム中にバグは残り続けてしまう．この問題を解決するために 90 年代半頃よりコードクローンの自動検出について研究が行われるようになった．現在までにさまざま検出法が提案されている．
コードクローン検出法は，前述の漏れのない修正支援以外でもさまざまな利用方法がある．例えば，ソースコードの重複率をメトリクスとして利用やコードクローン分布状態の可視化によるソースコードの特徴分析支援，将来の修正コストの削減を目的としたコードクローンのモジュール化・ライブラリ化支援，外注コードのチェック（不必要なコードの水増し，ライセンス違反のコード流用等），ソフトウェアシステム進化の分析手段等，である．しかし，論文等で利用の可能性が報告されるにとどまっており，実際に産業界での利用はあまり進んでいない．

２．研究の目的

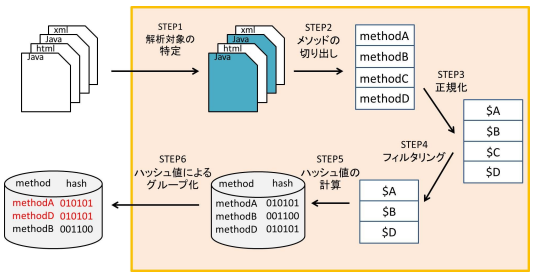
ソフトウェア中の重複コード（以下，コードクローン）を高速に，適切な粒度で検出する手法を確立し，システムとして構築する．１つのシステムの１つのリビジョンからの検出だけではなく，水平方向に巨大な対象（数千プロジェクト，10 億行程度）や，垂直方向に巨大な対象（10 万リビジョン程度の開発履歴）から，一台のワークステーションを用いて数時間程度でコードクローンの検出を完了できるシステムの構築を目指す．

３．研究の方法

本研究では，複数プロジェクトから高速にクローンを検出する手法および開発履歴からクローンを検出する手法を提案した．以下に各提案手法について述べる．

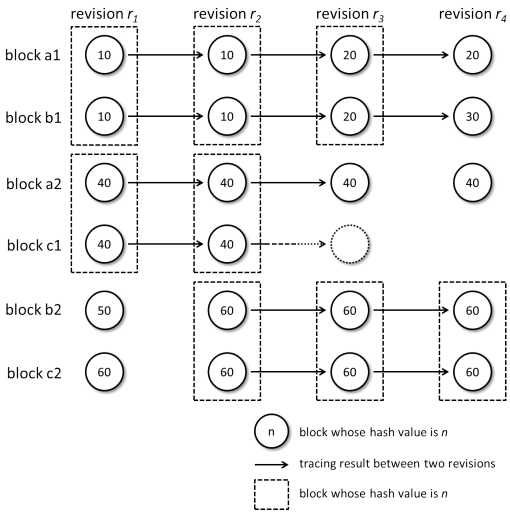
３．１．複数プロジェクトからのクローン検出

図１は提案手法の概要を表している．提案手法では，検出対象のソースコードから機能的なまとまりのモジュールであるメソッドや関数を抽出する．抽出した各メソッドについて，その中に含まれる字句列を基にしたハッシュ値を生成する．そして同じハッシュ値をもつメソッドをクローンとして検出する．この方法を用いることにより，従来の字句単位のクローン検出法に比べて劇的に高速化を実現できる．



３．２．複数リビジョンからのクローン検出

図２は提案手法の概要を表している．提案手法では，対象ソフトウェアの各リビジョンに対して，ソースコードに含まれる全てのブロック（クラス，メソッド，if 文や while 文等のブロック）を抽出する．そしてそれらに含まれる字句列を基にしたハッシュ値を生成する．ハッシュ値が同じブロックがクローンとみなされる．
また，既存手法の CRD という技術を利用する．CRD とは，コード片のシステム内での位置情報を表す表現形式である．本研究では CRD を利用することによって，クローンになっているコード片をリビジョンをまたいで追跡する技術も考案した．
ハッシュ値の計算は，各リビジョンで変更されたファイルに対してのみ計算を行うことにより，全てのリビジョンの全てのファイルに対して行う場合に比べて劇的に効率的に行うことができるよう実装を工夫した．



４．研究成果

ここでは各提案手法の成果について述べる．

４．１複数プロジェクトからのクローン検出

提案手法を実装したツールを UCI データセットという巨大なソースコードのデータセットに対して適用した．ソフトウェア数は約 13000，ファイル数は約 400 万，総行数は，約 3 億 6 千万行のデータセットである．実験には市販のパーソナルワークステーションを用いた．
検出の結果，対象メソッドのうち，約 52%がクローンになっていることがわかった．また従来のファイル単位では見つけることのできないクローンも多数存在していることがわかった．また全体のメソッドの 36%は他のソフトウェアのメソッドとクローンになっており，ソフトウェア間のクローンが非常に多く存在していることもわかった．
また検出に必要であった時間は約 2 時間であり，提案手法の高速に検出することを確認することができた．

４．２．複数リビジョンからのクローン検出

提案手法を複数のオープンソースソフトウェアに対して適用した．その結果，従来手法ではうまく検出および追跡できなかったが提案手法ではうまくできたクローンが数百あることが確認できた．
また実行時間は 2 時間程度であり，対象プログラムの規模を考慮すると高速に検出を終えることができたといえる結果であった．

５．主な発表論文等
（研究代表者、研究分担者及び連携研究者には下線）

〔雑誌論文〕（計 10 件）（全て査読有り）

[1] 石原知也，肥後芳樹，楠本真二，“書き忘れたコードに対するコード補完について”，電子情報通信学会論文誌 D，Vol.99-D，No.4，pp.415-427，2016 年 4 月．

[2] Jiachen Yang, Keisuke Hotta, Yoshiki Higo, Hiroshi Igaki, and Shinji Kusumoto, “Classification Model for Code Clones Based on Machine Learning”, Journal of Empirical Software Engineering (ESE), Volume 20, Issue 4, pp.1095-1125, August 2015.

[3] 今里文香，堀田圭佑，肥後芳樹，楠本真二，“機械学習を利用した危険なコードクローンの予測手法”，電子情報通信学会論文誌 D，Vol.J98-D，No.5，pp.847-850，2015 年 5 月．

[4] 堀田圭佑，楊嘉晨，肥後芳樹，楠本真二，“粗粒度なコードクローン検出手法の精度に関する調査”，情報処理学会論文誌，Vol.56，No.2，pp.580-592，2015 年 2 月．

[5] 村上寛明，肥後芳樹，楠本真二，“ギャップの位置情報を追加した正解クローンの生成”，電子情報通信学会論文誌 D，Vol.J97-D，No.8，pp.1537-1540，2014 年 9 月．

[6] 堀田圭佑，肥後芳樹，楠本真二，“CRD を用いたコードクローンの生存期間と修正回数に関する調査”，情報処理学会論文誌，Vol.55，No.2，pp.947-958，2014 年 2 月．

[7] 村上寛明，堀田圭佑，肥後芳樹，井垣宏，楠本真二，“Smith-Waterman アルゴリズムを利用したギャップを含むコードクローン検出”，情報処理学会論文誌，Vol.55，No.2，pp.981-993，2014 年 2 月．

[8] 長瀬義大，石原知也，楊嘉晨，堀田圭佑，肥後芳樹，井垣宏，楠本真二，“主要処理に着目したメソッド単位のコードクローン検出”，電子情報通信学会論文誌 D，Vol.J96-D，No.11，pp.2669-2680，2013 年 11 月．

[9] 村上寛明，堀田圭佑，肥後芳樹，井垣宏，楠本真二，“ソースコード中の繰り返し部分に着目したコードクローン検出ツールの実装と評価”，情報処理学会論文誌，Vol.54，No.2，pp.845-856，2013 年 2 月．

[10] 石原知也，堀田圭佑，肥後芳樹，井垣宏，楠本真二，“大規模なソフトウェア群を対象とするメソッド単位でのコードクローン検出”，情報処理学会論文誌，Vol.54，No.2，pp.835-844，2013 年 2 月．

〔学会発表〕（計 17 件）（全て査読有り）

[1] Yusuke Sabi, Hiroaki Murakami, Yoshiki Higo, and Shinji Kusumoto, “Reordering Results of Keyword-based Code Search for Supporting Simultaneous Code Changes”, In Proc. of the 23rd IEEE International Conference on Program Comprehension (ICPC2015), pp.289-290, Florence, Italy, May 18-19, 2015.

[2] Akio Ohtani, Yoshiki Higo, Tomoya Ishihara, and Shinji Kusumoto, “On the Level of Code Suggestion for Reuse”, In Proc. of the 9th International Workshop on Software Clones (IWSC2015), pp.26-32,

Montreal, Canada, March 6, 2015.

[3] Hiroaki Murakami, Yoshiki Higo, and Shinji Kusumoto, “ClonePacker: a Tool for Clone Set Visualization”, In Proc. of the 22nd IEEE International Conference on Software Analysis, Evolution, and Reengineering (SANER2015), pp.474-478, Montreal, Canada, March 2-6, 2015.

[4] Tomoya Ishihara, Yoshiki Higo, and Shinji Kusumoto, “How Often Are Necessary Code Missing? – A Controlled Experiment –“, In Proc. of the 14th International Conference on Software Reuse (ICSR2015), pp.156-163, Miami, Florida, USA, January 4-6, 2015.

[5] Ayaka Imazato, Keisuke Hotta, Yoshiki Higo, and Shinji Kusumoto, “Predicting Risky Clones Based on Machine Learning”, In Proc. of the 15th International Conference of Product Focused Software Development and Process Improvement (PROFES2014), pp.294-297, Helsinki, Finland, December 10-12, 2014.

[6] Yoshiki Higo, Shinji Kusumoto, “How Should We Measure Functional Sameness from Program Source Code? – An Exploratory Study on Java Methods –“, In Proc. of the 22nd International Symposium on the Foundations of Software Engineering (FSE2014), pp.294-305, Hong Kong, November 16-22, 2014.

[7] Hiroaki Murakami, Yoshiki Higo, Shinji Kusumoto, “A Dataset of Clone References with Gaps”, In Proc. of the 11th Working Conference on Mining Software Repositories (MSR2014), pp.412-415, Hyderabad, India. May 31 – June 1, 2014.

[8] Keisuke Hotta, Yoshiki Higo, and Shinji Kusumoto, “How Accurate Is Coarse-grained Clone Detection?: Comparison with Fine-grained Detectors”, In Proc. of the 8th International Workshop of Software Clones (IWSC2014), pp.1-18, Antwerp, Belgium, February 3, 2014.

[9] Tomoya Ishihara, Yoshiki Higo, and Shinji Kusumoto, “Reusing Reused Code”, In Proc. of the 20th Working Conference on Reverse Engineering (WCRE2013), pp.457-461, Koblenz, Germany, October 14-17, 2013.

[10] Yoshiki Higo, and Shinji Kusumoto, “Identifying Duplicate Code Removal Opportunities Based on Co-evolution Analysis”, In Proc. of the 13th International Workshop on Principles of Software Evolution (IWPSE2013), pp.63-67, Saint Petersburg, Russia, August 19-20, 2013.

[11] Yoshiki Higo, Keisuke Hotta, and Shinji Kusumoto, “Enhancement of CRD-based Clone Tracking”, In Proc. of the 13th International Workshop on Principles of Software Evolution (IWPSE2013), pp.28-37, Saint Petersburg, Russia, August 19-20, 2013.

[12] Hiroaki Murakami, Keisuke Hotta, Yoshiki Higo, Hiroshi Igaki, and Shinji Kusumoto, “Gapped Code Clone Detection with Lightweight Source Code Analysis”, In Proc. of the 21st International Conference on Program Comprehension (ICPC2013), pp.93-102, San Francisco, California, May 20-21, 2013.

[13] Tomoya Ishihara, Keisuke Hotta, Yoshiki Higo, Hiroshi Igaki, and Shinji Kusumoto, “Inter-Project Functional Clone Detection toward Building Libraries - An Empirical Study on 13,000 Projects -“, In Proc. of 19th Working Conference on Reverse Engineering (WCRE2012), pages 387-391, Canada, Kingston, October 15-18, 2012.

[14] Hiroaki Murakami, Keisuke Hotta, Yoshiki Higo, Hiroshi Igaki and Shinji Kusumoto, “Folding Repeated Instructions for Improving Token-Based Code Clone Detection”, In Proc. of 12th International Working Conference on Source Code Analysis and Manipulation (SCAM2012), pages 64-73, Italy, Riva del Garda, September 23-24, 2012.

〔図書〕（計 0 件）

〔産業財産権〕
○出願状況（計 0 件）

○取得状況（計 0 件）

〔その他〕
ホームページ等
https://github.com/YoshikiHigo/CloneGear/tree/master/CloneGear

６．研究組織
(1)研究代表者
肥後 芳樹（HIGO, Yoshiki）

大阪大学・大学院情報科学研究科・准教授
研究者番号：７０４５２４１４

(2)研究分担者
なし

(3)連携研究者
なし