テンプレートマッチングを用いた学習進捗表示システムの開発

馬場　雄佑[1]　山崎　進[1]

概要：本研究では，小学生にも扱えるビジュアルプログラミングツールでのプログラミング授業を想定し，その学習進捗を把握できるシステムの開発を行った．本研究ではこのシステムを，画像処理ライブラリである OpenCV によるテンプレートマッチング法と，API や機能が豊富なオンラインストレージである GoogleDrive を用いて開発し，学習進捗の数値化・表示が可能となった．また，2 種類のプログラミングツールでプログラミングを行い，開発したシステムを用いることで実験を行った．実験結果から，類似度の遷移によってプログラム作成の進行具合がある程度把握できること，テンプレート画像の大きさや背景によって類似度の範囲が変わるため，しきい値を動的に変化させる機能や背景を除去する機能などが必要であると考えられること，入力ミスなどを許す，つまりブロックの配置のみの判定であれば，開発したシステムで複数のツールでの判定が可能であることがわかった．

キーワード：テンプレートマッチング，ビジュアルプログラミング，プログラミング教育

Development of a Visualization System of Learning Progress Using Template Matching

Yusuke Baba[1]　Susumu Yamazaki[1]

1. はじめに

2017 年 3 月の学習指導要領の改定を受け，我が国でも，2020 年度から小学校でプログラミング教育が必修化される．プログラミングに取り組むことで，論理的思考力を養い，次世代の IT 人材として早期から育成することが目的である．しかし，プログラミング教育の内容や教え方は学校・教員の裁量とされており，また，研修や設備の整備は不十分であり，これによって小学校教師の授業の負担が大きくなっていくことが問題になっている．そこで，小学生にも扱えるビジュアルプログラミングツールでのプログラミング授業を想定し，その学習進捗を把握できるシステムを開発することで，この問題を解決できないかと考えた．本研究ではこのシステムを，画像処理ライブラリである OpenCV によるテンプレートマッチング法と，API や機能が豊富なオンラインストレージである GoogleDrive を用いて開発した．

[1] 北九州市立大学
The University of Kitakyushu

2. 既存研究

参考文献 [1] では，ウェブ上で動作するオンラインエディタとコーディング過程ビューで構成され，プログラミング演習におけるコーディング過程を記録し，可視化して講師に提示するシステムの提案をし，実際に Java プログラミング演習において実験を行っている．このシステムでは，オンラインエディタを用いることで受講生のコーディングプロセスにおける，文字入力，コンパイル，実行，提出といったすべての行動を記録している．このように，出力結果などのテキストを元に解析を行う研究は多く行われているが，ビジュアルプログラミングツールを用いた授業の場合，ツール内部の解析を行う必要があり，また，ツールに特化したシステムとなってしまう．本研究で開発するシステムは画像処理を用いるためツール内部の処理を解析する

必要がなく，様々なツールへの応用が期待できると考えられる．

3. OpenCV

参考文献 [2] によると，OpenCV（Open Source Computer Vision Liblary）とは，Intel，そして Willow Garage が開発・公開しているオープンソースのコンピュータビジョン用ライブラリである．本研究では，OpenCV によるテンプレートマッチング法を用いる．

4. テンプレートマッチング法

テンプレートマッチング法とは，特定の物体が存在するか，また，その物体がどこに位置しているかを調べる方法の一つである．まず，テンプレート画像と呼ばれる，探索したい物体の画像をあらかじめ用意する．そして，探索させたい入力画面内でテンプレート画像を移動させながら，入力画像とテンプレート画像との類似性を調べることで，入力画像中にテンプレート画像と同じパターンが存在するかどうかを調べる．

5. システムの構成

開発したシステムの構成を，図 1 に示す．

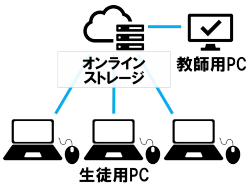

図 1 システムの構成

このシステムの構成は生徒用 PC，オンラインストレージ，教師用 PC の 3 つで構成されている．生徒用 PC では，生徒の操作している画面のスクリーンショット画像の作成・保存，そして OpenCV を用いたテンプレートマッチングをリアルタイムで一定間隔で行う．オンラインストレージでは，テンプレート画像の保存と，生徒用 PC で行ったマッチング結果を逐次表形式で編集・保存を行う．教師用 PC では，オンラインストレージに保存されているマッチング結果の表の表示を行う．

6. 動作手順

(1) オンラインストレージ上で，起動させた順に番号と進捗度の初期値 0 が割り当てる
(2) オンラインストレージから各生徒用 PC に，テンプレート画像をダウンロードさせる
(3) 各生徒用 PC 上で画面のスクリーンショット画像の作成とテンプレートマッチングを一定間隔で繰り返し行う
(4) オンラインストレージ上で進捗度を記録，図 2 のように表示する

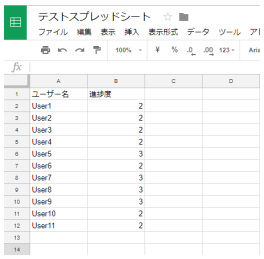

図 2 進捗度表示例

7. 実験内容

Scratch（参考文献 [4]，以下ビジュアルプログラミング言語 A）とプログラミン（参考文献 [5]，以下ビジュアルプログラミング言語 B）を学習対象として，以下の実験を行った．

(1) ビジュアルプログラミング言語 A とビジュアルプログラミング言語 B のそれぞれで，解答となるプログラムの画像を 2 パターン作成し，それぞれ template1，template2 とする．

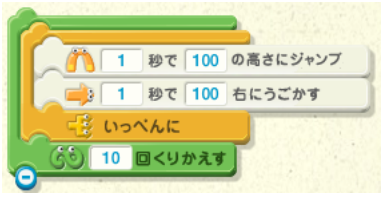

図 3 template1 の例

図 4 template2 の例

(2) 1 で作成した画像をテンプレート画像とし，実際にビジュアルプログラミング言語を用いて以下の手順でプログラムを作成し，開発したシステムを用いて類似度を計測する．
(a) ブロックを配置する
(b) ブロック内の入力を変更する
(c) プログラムが完成したらプログラムを実行する

(3)　上記をビジュアルプログラミング言語 A・B でそれぞれ 3 パターン行い，類似度の遷移と分布をまとめる．ビジュアルプログラミング言語 A を用いた実験を実験 1〜3，ビジュアルプログラミング言語 B を用いた実験を実験 4〜6 とする．

8.　実験結果

各実験における類似度の遷移を図 5〜図 10 に示す．

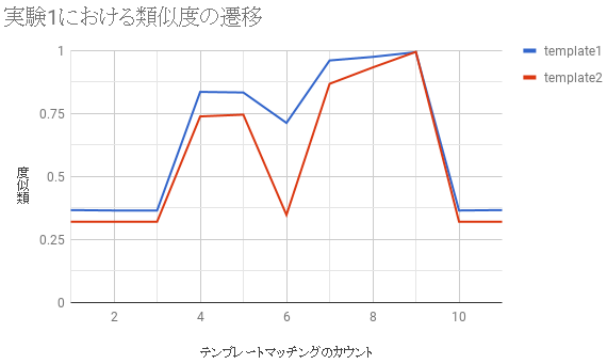

図 5　実験 1 における類似度の遷移

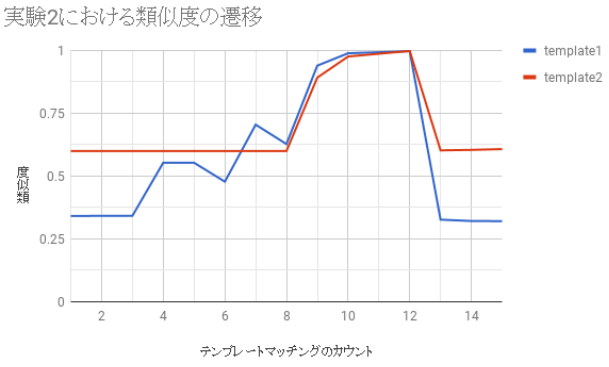

図 6　実験 2 における類似度の遷移

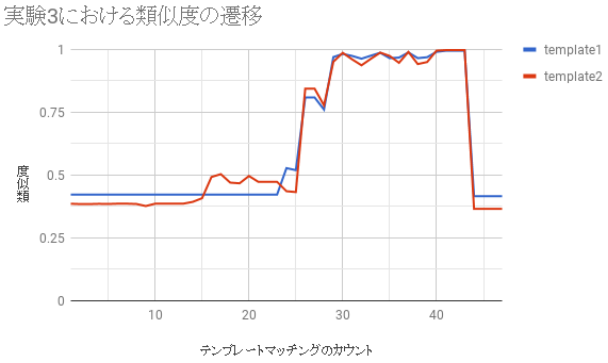

図 7　実験 3 における類似度の遷移

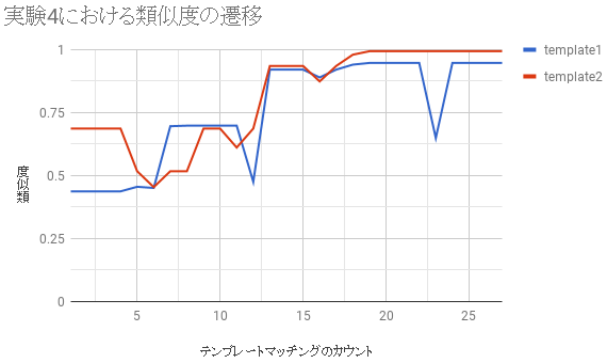

図 8　実験 4 における類似度の遷移

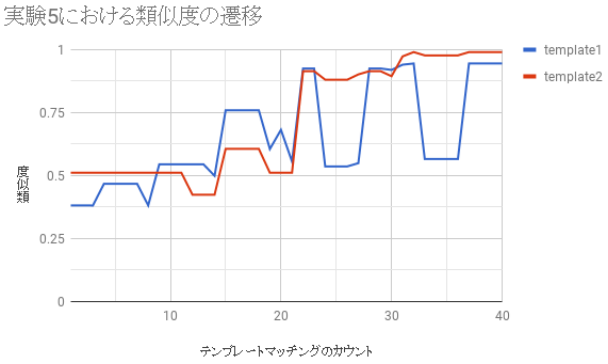

図 9　実験 5 における類似度の遷移

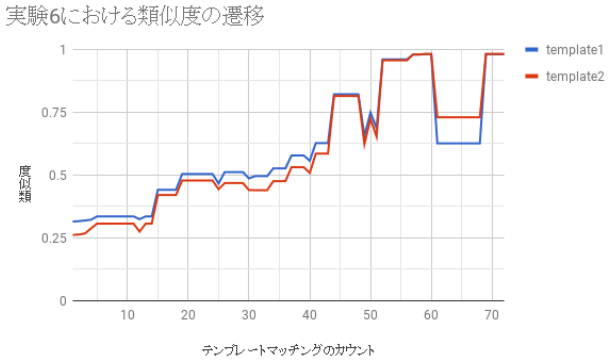

図 10 実験 6 における類似度の遷移

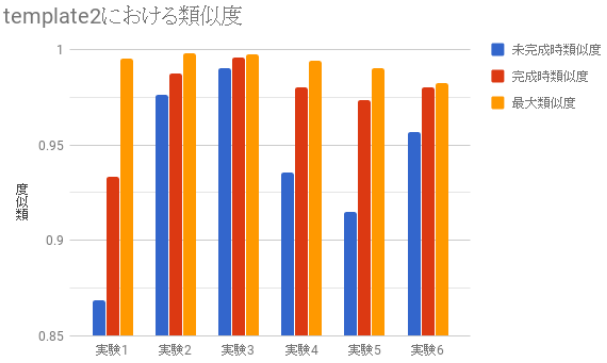

図 12 template2 における類似度

各テンプレート画像における類似度を，プログラム完成前の最大類似度を未完成時類似度，プログラム完成時の類似度を完成時類似度，プログラム完成後の最大類似度を最大類似度としてまとめたものを，図 11〜図 12，表 1〜表 2 に示す．また，ブロックの配置に注目し，各テンプレート画像における類似度を，ブロックの配置完成前の最大類似度を配置未完成時類似度，ブロックの配置完成時の類似度を配置完成時類似度，プログラム完成後の最大類似度を最大類似度として，図 13〜図 14，表 3〜表 4 に示す．

表 1 template1 における各実験の類似度

	未完成時類似度	完成時類似度	最大類似度
実験 1	0.96121281	0.97541696	0.99407715
実験 2	0.98908091	0.99309391	0.99726105
実験 3	0.98833323	0.99075258	0.99515283
実験 4	0.92094761	0.94067097	0.94735944
実験 5	0.92544019	0.94048440	0.94522542
実験 6	0.96087843	0.97978556	0.98158634

表 2 template2 における各実験の類似度

	未完成時類似度	完成時類似度	最大類似度
実験 1	0.86850286	0.93349630	0.99536752
実験 2	0.97604972	0.98726368	0.99816626
実験 3	0.99043399	0.99563348	0.99761260
実験 4	0.93553573	0.98015565	0.99383771
実験 5	0.91484654	0.97337598	0.99047333
実験 6	0.95679814	0.98005414	0.98225504

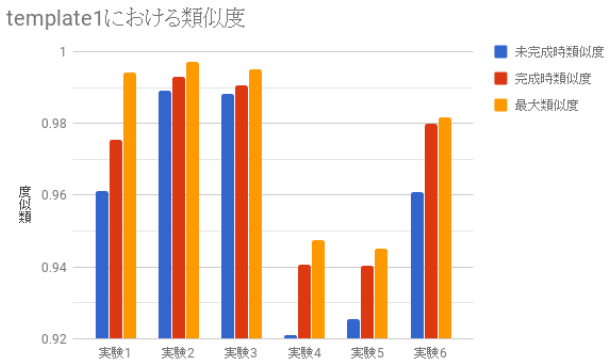

図 11 template1 における類似度

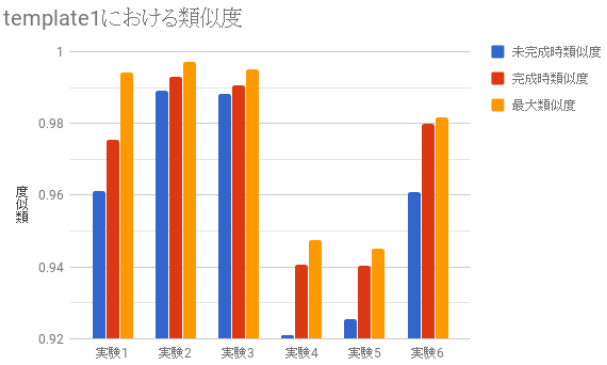

図 13 ブロックの配置に注目した template1 における類似度

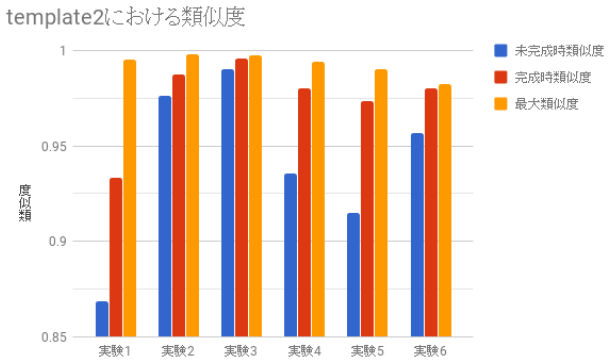

図 14 ブロックの配置に注目した template2 における類似度

表 3 ブロックの配置に注目した template1 における各実験の類似度

	配置未完成時類似度	配置完成時類似度	最大類似度
実験 1	0.36732623	0.83641487	0.99407715
実験 2	0.70496899	0.93977970	0.99726105
実験 3	0.80925947	0.96993023	0.99515283
実験 4	0.69844407	0.92094761	0.94735944
実験 5	0.75948101	0.92544019	0.94522542
実験 6	0.82180732	0.96087843	0.98158634

表 4 ブロックの配置に注目した template2 における各実験の類似度

	配置未完成時類似度	配置完成時類似度	最大類似度
実験 1	0.32094553	0.73907197	0.99536753
実験 2	0.60004342	0.89216918	0.99816626
実験 3	0.84421015	0.95126021	0.99761260
実験 4	0.68695861	0.93553573	0.99383771
実験 5	0.60632175	0.91484654	0.99047333
実験 6	0.81500554	0.95679814	0.98225504

9. 考察

図 5〜図 10 から，どのパターンにおいても，類似度の遷移によってプログラム作成の進行具合がある程度把握できる事が分かる．また，本研究における実験では，プログラムの作成の進行具合は常に完成へ近づくように操作しているにもかかわらず，どのパターンにおいても，類似度の値が下がるタイミングがあることがわかる．このタイミング周辺のスクリーンショット画像から，この現象が起こる原因は，ブロックを配置し終わった時点で一度類似度が高くなるが，その後ブロック内の入力を行う際にカーソルやマウスポインタがブロック上に重なり，プログラムの内容が正しくスクリーンショット画像に反映されないことであるとわかった．

次に，図 11〜図 12 から，なるべくブロックの判定に必要な範囲のみになるように作成したテンプレート画像である template2 の方が，ひとつのプログラム分全体を含む template1 よりも類似度の値の分布が広いことが分かった．これは template2 の方が画像が小さいため，ブロック内の入力文字の違いによる類似度への影響が大きいこと，ビジュアルプログラミング言語 A・B のどちらも，背景に模様があるため，プログラム自体の配置によって背景が変化し，それが類似度へ影響してしまうことが原因であると考えられる．また，図 10 からわかる通り，実験 6 においては template1 と template2 による違いはあまり見られない．これは一部の大きなブロックによって判定に必要な範囲が広くなってしまったためである．これを解決するためにはテンプレートマッチングの前処理として判定に必要ない背景部分をなくすなど，システムの改良が必要であると考えられる．

さらに，図 11〜図 12，表 1〜表 2 から，本実験における類似度のしきい値について考える．template1，template2 のどちらを用いた場合でも，実験 1〜3 における不正解であるプログラムに対する類似度の最大値が，実験 4〜6 における最大類似度をこえてしまっていることが分かる．つまり，実験 1〜3 のパターンに合わせてしきい値を設定した場合，実験 4〜6 のパターンに対して正解判定を出すことが出来なくなる．逆に実験 4〜5 に合わせた場合は，実験 1〜3 において実際は不正解であるプログラムに対して正解判定を出してしまうことになる．よって，しきい値の値を正確に設定するためには，前述したようなシステムの改良や処理前の画像の大きさなどによってしきい値を動的に変化させる機能が必要である．また，図 5〜図 10 から，どのパターンにおいても，ブロックの配置が正しくなった時点で類似度が大きく変化することから，ブロックの配置に注目して考える．図 13〜図 14，表 3〜表 4 から，実験におけるすべてのパターンの配置完成時類似度の最大値から実験におけるすべてのパターンの最大類似度の最小値，つまり，0.84421015〜0.94522542 の間にしきい値を設定することで，「少なくともブロックの配置は終わった」という判断が出来るということが分かる．よって，入力ミスなどを許す，つまりブロックの配置のみの判定であれば開発したシステムでも十分判定が可能であると考えられる．

10. おわりに

本研究では，学習者の PC 画面のスクリーンショット画像を元にテンプレートマッチングを行い，その結果を学習進捗として数値化・表示をするシステムを開発し，その性能の一部を実験により評価した．実験結果から，類似度の遷移によってプログラム作成の進行具合がある程度把握

できること，テンプレート画像の大きさや輝度によって類似度範囲が変わるため，画像によってしきい値を動的に変化させる機能が有効であると考えられること，入力ミスなどを許す，つまりブロックの配置のみの判定であれば開発したシステムで十分判定が可能であることがわかった．また将来課題として，テンプレート画像を自動で作成するシステムの作成，また，実際にシステムを用いて授業を行い，教師側，学習者側それぞれにおいてユーザビリティテストを行うことなどが挙げられる．

謝辞　研究におけるご指導をしていただきました北九州市立大学国際環境工学部情報メディア工学科の山崎進准教授に感謝の意を表します．また，研究にご協力いただいた株式会社システムトランジスタの皆様，飯塚市立片島小学校の皆様，山崎進研究室の皆様に感謝の意を表します．

参考文献

[1] 井垣 宏ほか，「プログラミング演習における進捗状況把握のためのコーディング過程可視化システム C3PV の提案」，『情報処理学会論文誌』，Vol.54　No.1 330339，2013
[2] Gary Bradski・Adrian Kaehler(訳:松田晃一)，詳解 OpenCV コンピュータビジョンライブラリを使った画像処理・認識，株式会社オライリー・ジャパン，2009
[3] 奈良先端科学技術大学院大学 OpenCV プログラミングブック制作チーム，OpenCV プログラミングブック，株式会社毎日コミュニケーションズ，2007
[4] 文部科学省，プログラミン，http://www.mext.go.jp/programin/
[5] MIT メディアラボ ライフロングキンダーガーテングループ，Scratch，https://scratch.mit.edu/