

Scratch における類似動作を含む作品間の プログラム類似性の分析

三倉 舞子† 伊原 彰紀† 橋谷 直樹†
Mikura Maiko Akinori Ihara Naoki Hashitani

1. はじめに

初等教育からのプログラミング必修化に伴い、ビジュアルプログラミング言語を利用した教育が進められている。Java 言語や C 言語のようなテキストで記述するプログラミングとは異なり、ビジュアルプログラミング言語は命令機能を持つオブジェクトの組み合わせによって視覚的にプログラミングを行う言語である。このため、プログラミング初学者は直感的にプログラムを実装でき、プログラミングを学ぶことができる。ブロックタイプのビジュアルプログラミングの例として Scratch がある。Scratch は、ユーザが作成したプログラム作品を Scratch サービス上で公開することができ、ユーザは他作品を見ることで知見を広げることができる。

ビジュアルプログラミングでは、学習者が自由な発想で作品を制作するため、既存のプログラムを模倣することから学習を開始する。しかし、指導者から与えられた課題、または学習者自身が作成しようとする作品のために参考となる作品の検索は容易ではない。Scratch では、作品を探す検索機能を提供している。しかし、2022 年 7 月 18 日現在において、Scratch は 1 億作品以上を公開しているため、膨大な検索結果の中からユーザが参考にする作品を見つけることは容易ではない。特に、ビジュアルプログラミングは、限られたブロックを組み合わせ、使用するブロックが少数の作品も多いため、実装方法が類似しているプログラム作品が多数存在する。検索結果の作品のプログラムを 1 つ 1 つが確認することは、多様な実装を学ぶ利用者にとって効率的な方法ではない。

本研究では、構造的に類似するプログラムを有するビジュアルプログラミング作品を分類し、多様なプログラム作品を効率的に収集する手法を提案する。本研究では、Scratch 作品のプログラムを、プログラム解析で使用される抽象構文木 (AST) に変換し、木構造間の編集距離を求めることで、類似するプログラムの収集を目指す。本手法の評価では、オブジェクトの移動軌跡が類似するプログラムを有する作品を対象に、提案手法による作品分類の課題を明らかにする。

続く 2 章では Scratch を用いたプログラミング学習の背景と従来研究を述べる。3 章では、提案手法について説明し、4 章でケーススタディと結果について述べる。5 章ではケーススタディの考察を行い、最後に 6 章でまとめと今後の課題を述べる。

2. ビジュアルプログラミングにおける検索

2.1 Scratch

Scratch は、MIT メディアラボが開発するビジュアルプログラミング言語である。テキストプログラミングと異なり、命令処理を持つブロックを組み合わせ直感的にプログラミングを行うことができるため、世界中でプログラミング教育に用いられている。Scratch では、個々の命令処理を「ブロック」、ブロックを組み合わせ関数として実装したプログラムを「スクリプト」、1 つ以上のスクリプトによって動作が決定されるオブジェクトを「スプライト」と呼ばれる。1 つの作品には、1 つ以上のスプライトで構成される。ユーザが制作した作品は、Scratch サービス上に公開することができる。Scratch は、公開されている作品を複製する機能「リミックス」を提供しており、ユーザは他の作品を複製し、コードの追加や削除を行い、容易に新たな別の作品を制作することができる。

2.2 従来研究

ユーザは、Scratch オンラインサービス上に公開されている膨大な数の作品を参照することで多様な実装方法を学習することができる [1]。参照する作品を探し出すために、ユーザは作品のタイトルと説明文を対象としたキーワード検索を行う。検索結果としてブックマーク数などの人気の高い作品が順に提示される。しかし、ユーザがイメージする処理をテキストに変換することは困難であり、ユーザが参考にできる作品を見つけることは容易ではない [2]。

先行研究では、Scratch 作品の直感的な検索を目的として、ユーザがマウス等でオブジェクトの移動軌跡を入力し、その軌跡と類似する作品を収集する手法を提案した [3]。本手法では、動的時間伸縮法 (DTW: Dynamic time warping) を用いて、ユーザが入力した軌跡と Scratch で公開する作品中のオブジェクトの軌跡の類似性を計測している。DTW は、2 つの検索対象作品のオブジェクト動作として座標変化を比較し、座標の差分を距離として算出する。先行研究では、距離が 4 未満のスクリプト対は類似していることを実験的に明らかにしている。しかし、Aivaloglou らの調査 [4][5] では、Scratch で制作されるプログラム規模は比較的小さく、複雑度も単純であるため、先行研究においても類似するプログラムが多数確認しており、検索において多様なプログラムの収集は容易でない。

表1 分析対象外作品に使用されるブロック
動きブロック

マウスポインタへ向ける	マウスポインタへ行く
-------------	------------

イベントブロック

()キーが押されたとき	このスプライトがクリックされたとき
--------------	-------------------

調べるブロック

マウスポインタに触れた	マウスポインタまでの距離
()と聞いて待つ	()キーが押された
マウスが押された	マウスの X 座標
マウスの Y 座標	

表2 実装の種類

	制御文を使用	制御文を不使用
座標指定	4	1, 2, 6, 7
移動	3, 5, 8, 9, 10, 11	12, 14

3. 提案手法

3.1 概要

本研究は、多様な方法で実装された作品を効率的に収集することを目的とする。類似するプログラム構造を有する作品の収集方法を提案し、本手法の課題を明らかにする。

3.2 手法

3.2.1 収集したプログラムをASTに変換

Scratch プログラムを、ブロックのつながりを示す抽象構文木 (AST) は、プログラムの構文情報を木構造で表したものである。AST は順序木であり、木構造におけるノードはプログラムの構文情報、エッジで結ばれた子ノードは詳細情報を示す。本研究では、Scratch におけるブロックをノードとして木構造に変換する。ただし、ブロックが持つ変数の情報は持たないように実装する。

3.2.2 編集距離を用いて作品間の類似度を算出

ASTに変換したScratchプログラムの類似度を算出する。本研究ではキーワード検索で抽出される作品が持つプログラムの実装方法の類似性を評価するために Zhang らが提案する木構造間の編集距離手法を用いる。木データ T_1 と木データ T_2 の編集距離は、 T_1 を T_2 に変換するためにノード削除、ノード挿入、ノードラベル置換の 3 種類の編集操作の回数の最小値である。

3.2.3 類似度に基づく分類

Scratch 作品には、1 つ以上のスプライト、また、各スプライトには 1 つ以上のスクリプトで構成されている。本研究では、Scratch が提供する検索機能から収集した作品をスクリプトに分解し、全てのスクリプトを総当たりで編集距離を算出し、類似スクリプトを俯瞰的に可視化するネットワークグラフを作成する。ノードをスクリプト、エッジを編集距離の重みによって異なる太さで示す。本研究では、類似するとみなすことができるスクリプトを分類する編集距離の閾値を分析する。

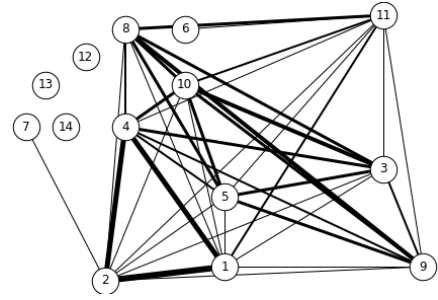


図1 スクリプト間の編集距離に基づくネットワークグラフ

1	0	1	6	2	6	9	7	6	6	6	5	33	16	14
2	1	0	6	2	6	10	6	6	6	6	6	34	16	15
3	6	6	0	4	4	10	11	4	5	3	6	34	20	15
4	2	2	4	0	5	11	7	5	5	4	7	35	16	16
5	6	6	4	5	0	10	11	4	4	4	6	32	20	14
6	9	10	10	11	10	0	10	9	9	10	8	26	12	9
7	7	6	11	7	11	10	0	11	11	11	11	32	20	15
8	6	6	4	5	4	9	11	0	3	4	4	32	20	13
9	6	6	5	5	4	9	11	3	0	4	6	31	19	12
10	6	6	3	4	4	10	11	4	4	0	5	34	20	15
11	5	6	6	7	6	8	11	4	6	5	0	31	19	12
12	33	34	34	35	32	26	32	32	31	34	31	0	28	20
13	16	16	20	16	20	12	20	20	19	20	19	28	0	14
14	14	15	15	16	14	9	15	13	12	15	12	20	14	0
1	2	3	4	5	6	7	8	9	10	11	12	13	14	14

図2 スクリプト間の編集距離

4. ケーススタディ

4.1 データセット

本研究では、Scratch のスクリプトが構造的に類似する作品を収集するために、スクリプト間の類似を判定する閾値を分析する。本研究では、構造的に類似する作品を分析するために、オブジェクトの移動軌跡が類似する作品のみを対象に、プログラムの構造的に類似するスクリプトを分析する。具体的には、キー入力やマウス操作を必要とするブロックを含む作品 (表 1)、Scratch が提供するオブジェクト以外を使用した作品を、Aivaloglou らが公開するデータセットから除いた。得られた 13,437 件の作品からオブジェクトが四角形に動作する作品を先行研究の手法により抽出した 14 件の作品を本研究の分析対象とする。

14 件のスクリプトは、著者らが目視によって類似するスクリプトを分類する。座標を指定しオブジェクトを移動するスクリプトと、オブジェクトを回転させるスクリプトの 2 種類、また、制御文として繰り返し命令処理を含むか否かによる 2 種類、合計 4 種類に分類する。表 2 は 14 件のスクリプトの分類を示す。本研究では、提案手法が 4 種類に分類できるか否か、また分類のための課題を明らかにする。

4.3 実験結果

4.3.1 編集距離の分析

図 1 は、14 件のスクリプト間の編集距離に基づくネットワークグラフを示す。ノードをスクリプト、エッジは編集距離が 6 以上の重みを有する場合のみ示す。また、図 2 は、スクリプト間の編集距離をヒートマップで示す。縦軸と横軸はスクリプトを識別するための ID を示す。色が濃いセル

ほど編集距離が短い、つまりスクリプト間の類似度が高いことを示す。

ID12, 13, 14 のスクリプトは、それぞれ使用されるブロック数が 35, 23, 16 であり、他のスクリプトに比べて多い (ID12, 13, 14 を除くスクリプトの平均ブロック数平均 6.9) 。分析対象の作品規模の分散が大きい場合に類似度の閾値の決定が困難になるため、本研究では、編集距離を編集距離の最大値で除算することで正規化する。ただし、本手法はブロック数が多いスクリプトは、ブロック数の少ないスクリプトと部分的に類似している場合、編集距離が大きくなってしまいう課題がある。ID4¹ と 13² は 13 が 4 を含む規模の大きいプログラムのため部分的に類似したスクリプトであるが、編集距離が大きくなる。今後は、類似しているスクリプトの一部分のみを抽出して評価する方法を検討する。

4.3.2 正規化した編集距離の分析

スクリプト間の編集距離を、分析対象としたスクリプト対の最大編集距離で除算することで正規化した結果と、4.1 項において目視で分類した結果をそれぞれ図 3 と図 4 に示す。図 3 と図 4 の横軸と縦軸はそれぞれ、図 2 と同様にスクリプトの ID を示す。色が濃いセルほど編集距離が短い、つまりスクリプト間の類似度が高いことを示す。図 4 は、類似する作品を白以外の異なる色で示し、先行研究により類似する作品として判定した編集距離が閾値 0.5 以下のスクリプトのみの値を表記する。

ID1, 2, 4 はいずれも座標指定することでオブジェクトが四角に移動するスクリプトであるため編集距離が短く、本手法では類似するスクリプトに分類される。しかし、ID4 は制御文として for を使用している一方で、ID1,2 は制御文を使用していない。制御文はプログラム構造の根幹となるため、今後は編集距離の算出において、ブロックの種類による編集の影響を検討する。

ID1, 2 と ID6, 7 は類似するスクリプトであるが、編集距離は長くなる。その理由は、ID1,2 は移動先の x 座標、y 座標を指定ブロックのみで構成しているのに対し、ID6, 7 は座標指定ブロックの間に「n 秒待つ」などのブロックが使用されているため編集距離が長くなった。同様の理由で ID8, 9, 10, 11 も構造に影響しないブロックの多用により、類似スクリプトにもかかわらず編集距離が長くなった。今後は、制御文にかかわらず、ブロックの種類による編集距離への影響を分析する。

本手法では編集距離の閾値を 0.5 として分析したが、類似する作品を取りこぼしていることが示唆されるため、今後の研究において閾値を検討する。

5. 妥当性への脅威

本研究の内的妥当性は、提案する編集距離がスクリプトに使用されるブロック数に影響を受けるため、大きく異なる規模のスクリプト間の編集距離が長くなる。本手法は、同じ規模のスクリプトの類似性を計測するためには有効である。今後は、スクリプト中の部分的な類似性を計測する方法を検討する。

本研究の外的妥当性は、ケーススタディとして対象とした作品は、オブジェクトの移動軌跡が類似する作品を対象

1	0	0.2	1	0.3	1	0.8	0.6	1	0.9	1	0.7	0.9	0.7	0.9
2	0.2	0	1	0.3	1	0.9	0.6	1	0.9	1	0.9	1	0.7	0.9
3	1	1	0	0.7	0.7	0.9	1	0.7	0.7	0.5	0.9	1	0.9	0.9
4	0.3	0.3	0.7	0	0.8	1	0.6	0.8	0.7	0.7	1	1	0.7	1
5	1	1	0.7	0.8	0	0.9	1	0.7	0.7	0.7	0.9	0.9	0.9	0.9
6	0.8	0.9	0.9	1	0.9	0	0.9	0.8	0.8	0.9	0.7	0.7	0.5	0.6
7	0.6	0.6	1	0.6	1	0.9	0	1	1	1	1	1	0.9	0.9
8	1	1	0.7	0.8	0.7	0.8	1	0	0.4	0.8	0.6	0.9	0.9	0.8
9	0.9	0.9	0.7	0.7	0.7	0.8	1	0.4	0	0.6	0.9	0.9	0.8	0.8
10	1	1	0.5	0.7	0.7	0.9	1	0.8	0.6	0	0.7	1	0.9	0.9
11	0.7	0.9	0.9	1	0.9	0.7	1	0.6	0.9	0.7	0	0.9	0.8	0.8
12	0.9	1	1	1	0.9	0.7	0.9	0.9	0.9	1	0.9	0	0.8	0.6
13	0.7	0.7	0.9	0.7	0.9	0.5	0.9	0.9	0.8	0.9	0.8	0.8	0	0.6
14	0.9	0.9	0.9	1	0.9	0.6	0.9	0.8	0.8	0.9	0.8	0.6	0.6	0
	1	2	3	4	5	6	7	8	9	10	11	12	13	14

図 3 標準化したスクリプト間の編集距離

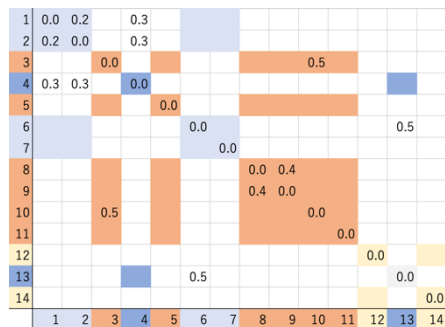


図 4 類似作品と正規化した編集距離との関係

としており、ユーザのキー入力やマウス操作を必要とするブロックを使用した作品は今後の分析対象とする。

6. おわりに

本研究では、ビジュアルプログラミング作品における多様なプログラム作品の効率的な収集に向けて、Scratch を対象に類似するプログラムを収集する手法を提案した。特に、オブジェクトの移動軌跡が類似するプログラムを有する作品を対象に、Scratch のプログラムを AST に変換し、木構造間の編集距離によって類似度を算出する方法の課題を明らかにした。Scratch でブロック数の少ない作品が制作されることが多いが、作品の規模によって編集距離の分散が大きくなるため、スクリプト中の部分的に類似する箇所を特定することが今後の課題である。また、スクリプトの構造の根幹に影響しない命令処理は、類似度の算出において対象外にする処理も必要であることを明らかにした。本手法では、編集距離によって一部のスクリプトの類似性を明らかにしたが、今後は編集距離算出の事前処理を検討し、類似する作品の収集精度の向上を目指す。

参考文献

- [1]. Mitchel Resnick, John Maloney, Andr es Monroy-Hern andez, Natalie Rusk, Evelyn Eastmond, Karen Brennan, Amon Millner, Eric Rosenbaum, Jay Saul Silver, Brian S Silverman, Yasmin Bettina Kafai, Scratch: Programming for all, *Communications of the ACM*, Vol.52, No.11, pp.60-67, (2009).
- [2]. Sheela Surisetty, Catherine Law, Chris Scaffidi, Behavior-based clustering of visual code, *Proceedings of IEEE Symposium on Visual Languages and Human-Centric Computing*, pp.261-269, (2015).

¹ <https://scratch.mit.edu/projects/98149958>

² <https://scratch.mit.edu/projects/99525787>

- [3]. 福地 ユキ, 伊原 彰紀, 山本 豪志朗, 橋谷 直樹.: ビジュアルプログラミング作品検索のためのオブジェクト操作データの時系列分析, 情報処理学会関西支部支部大会講演論文集, pp.1-6, (2021)
- [4]. Aivaloglou, E. and Hermans, F.: How Kids Code and How We Know: An Exploratory Study on the Scratch Repository, *Proceedings of the Conference on International Computing Education Research (ICER'16)*, pp.53–61 (2016)
- [5]. Efthimia Aivaloglou and Felienne Hermans, “How kids code and how we know: An exploratory study on the scratch repository,” *Proceedings of the Conference on International Computing Education Research (ICER'16)*, pp.53–61, (2016).