

# 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解析

福地 ユキ† 伊原 彰紀† 山本 豪志朗‡ 橋谷 直樹†  
Yuki Fukuchi Akinori Ihara Goshiro Yamamoto 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プログラミング教育が進められている。ビジュアルプログラミング言語は、プログラム中の「イベントを受け取る」、「命令を出す」、「値を返す」などの命令処理をブロックで視覚的に表現し、そのブロックを組み合わせることでプログラミングができる。ビジュアルプログラミング言語は、テキストプログラミングのようにプログラムの記法にとらわれることなく実装できる。ビジュアルプログラミングを用いた代表的な学習サービス Scratch や Blockly などでは、作成したプログラム作品を個々のサービス上で公開しており、学習者は公開された作品を参照しながら作成することができる。

ビジュアルプログラミング言語では、数学や英語のような明確な学習順序がなく、学習者が自由な発想で作品を制作しており、公開されている既存のプログラムを模倣することで学習することも多い。学習者にとって参考になり得る作品は、各サービスが提供する検索エンジンにおいて、自然言語の単語を使った検索によって発見することができる。しかし、学習者がイメージする命令処理を言語化することは困難であるため、学習者が制作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。学習者のイメージをプログラムの論理的な命令処理へ変換するための研究や技術は現時点で存在しないため、学習者にとって論理的な命令処理へ変換は、ビジュアルプログラミング学習を進める障壁になると考える。

本研究では、学習者が制作しようとする命令処理（特に、キャラクタの動作）を視覚的なオブジェクト操作として表現し、そのオブジェクト操作を検索クエリとして用いた検索手法を提案する。具体的には、オブジェクトの移動をスナップショット画像として保存し、サービスに公開される膨大なプログラム作品の中から入力と類似するオブジェクト動作を有する作品を DTW (Dynamic Time Warping, 動的時間伸縮法) を用いて収集する。本研究では、多様な入力を検討するために、Scratch において公開される作品 95,432 件間で類似するオブジェクト動作を有する作品の分類を行い、本提案手法の適用可能性を検証する。

テキストベースのプログラムでは、命令処理の完全一致によりプログラム検索手法が提案されているが、ビジュアルプログラミングを対象に画像検索することで、実装方法の異なる類似作品の検出が期待できる。

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 章で実験と結果について述べる。5 章では実験結果の考察し、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しているビジュアルプログラミング言語の 1 つであり、命令処理を視覚的に表現したブロックを組み合わせることで直感的なプログラミングを実現している。テキストベースのプログラミング言語を使用した学習と比較して、Scratch は文字入力の間違いによるエラーが発生しないため学習難易度が低く、プログラミング初学者の学習ツールとして利用されることが多い。従来研究では、Scratch を用いた学習がプログラミング学習に有効であり、テキストベースのプログラミングへの移行を容易にすることを実証実験により明らかにしている。学習者は制作した作品を、Scratch サービス上に公開することができる。また、Scratch で公開されている作品を複製し、再利用する機能として「リミックス」が提供されており、学習者はリミックスにより他の作品を複製し、コードの追加や削除を行うことで新たな別の作品として制作することが可能である。

### 2.2 Scratch における検索手順

Scratch では、サービスに公開された作品を検索する機能を有する。Scratch 作品の検索では、自然言語の単語を検索クエリとして入力し、作品のタイトルと説明文に検索単語を含む作品が出力される。検索対象となった作品は、人気の高い作品が検索上位に表示される。人気のある作品の中には多数の命令処理や複雑な命令処理を含む作品も多いため、プログラミング初学者にとって参考になる作品とは限らない。

C 言語や Java 言語をはじめとするテキストプログラミング言語では、メソッド名などの単語を使ったプログラム検索の研究が多数行われている [1]。しかし、ビジュアルプログラミング言語においては、実装内容を自然言語の単語で検索することは不向きであり、特に論理的な命令処理に慣れていないプログラミング初学者が検索することは容易ではない。本論文では、学習者のイメージを視覚的なオブジェクトの操作で表現し、類似する命令処理を含む作品を検索する手法を提案する。

### 2.3 従来研究

CT (Computational Thinking) スキルを育むためにはビジュアルプログラミング言語を用いた教育が有効である [2]。特に、Scratch は優れたプログラミング学習環境として、世

†和歌山大学, Wakayama University

‡京都大学附属病院, Kyoto University Hospital

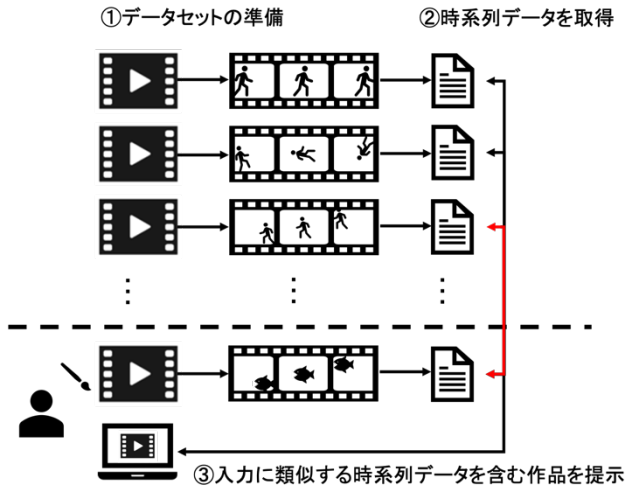


図1 提案手法の流れ

界中の教育機関で利用されており，Scratch の学習効果に関する研究が進められている[3] [4] [5] [6] [7][8].

Scratch で制作された作品の特徴を調査するために，Aivaloglou らは Scratch における学習者 109,960 人が制作した公開作品 250,163 件から，各作品に含まれるプログラム，および各作品を Dr.Scratch によって評価した結果を収集し，公開している[9]. Aivaloglou らは実行可能なプログラムを含む作品 233,491 件を解析した結果，自身で制作する作品のプログラム規模は小さく，複雑度も単純であった[9]. Dasgupta らは，Scratch において学習者がリミックスを使用して作品を制作することで，多様な命令処理を使用した作品を制作できるようになることを明らかにした [6].

### 3. 提案手法

本章では，学習者がイメージする命令処理を視覚的なオブジェクト操作として表現し，そのオブジェクト操作を検索クエリとして用いた検索手法を述べる．

#### 3.1 概要

本手法は，データベースとして事前にビジュアルプログラミングサービスの各作品中に登場するオブジェクトの移動方向，距離を収集し，学習者がイメージする処理と類似する作品をから検出する．図 1 は，本提案手法の概略図を示す．

##### 3.1.1 SIFT 特徴量

オブジェクトの移動方向，距離の収集には SIFT 特徴量 [10]を用いる．SIFT 特徴量は，画像認識や画像中の特徴点の追跡に局所特徴量を利用する手法である．

##### 3.1.2 動的時間伸縮法 (DTW: Dynamic time warping)

検索するオブジェクトの移動とデータベースにおける作品中のオブジェクトの移動との類似度の算出には動的時間伸縮法 (DTW) を用いる．DTW は，時系列データの距離を測る際に用いる手法である．具体的には DTW は 2 つの時系列の各点の距離 (誤差の絶対値) を総当たりで算出し，全て求めた上で 2 つの時系列が最短となる経路を見つける．例えば，図 2 に示すように類似度を算出した 2 つの時系列データ  $S=s_1, s_2, \dots, s_m, T=t_1, t_2, \dots, t_n$  を各時系列データを横と縦に並べ，点  $(i, j)$  における要素  $s_i$  と  $t_j$  間の距離を算出

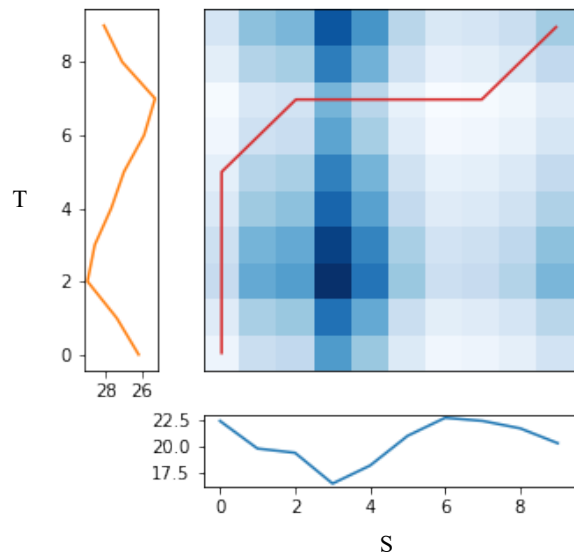


図2 DTW の概念図

する．時系列データ S と T の各要素の距離が最も小さくなる点の系列から類似度を算出する手法である．本研究では，DTW を用いて，学習者がイメージするオブジェクトの移動情報と類似するオブジェクト移動を含む作品を検索する．

#### 3.2 ビジュアルプログラム作品の収集

検索対象とするビジュアルプログラム作品を収集し，作品中で使用されるオブジェクトの移動方向，移動距離を収集する．

##### 3.2.1 作品収集

ビジュアルプログラミングサービスにおいて公開する各作品の時系列フレーム画像を収集する．フレーム画像の収集は，Web アプリケーションの自動テストツール Selenium を用いて，検索対象作品の実行画面から 0.3 秒に 1 枚のスナップショットを取得する．実行が終わらず無限ループする作品に対しては最大 100 枚のスナップショット画像を収集する．

##### 3.2.2 作品中のオブジェクト画像の位置情報を計測

ビジュアルプログラミング作品のオブジェクト画像を事前に収集し，そのオブジェクト画像をテンプレート画像として，SIFT 特徴量を用いて 3.2.1 で収集した各スナップショット画像の中からテンプレート画像と同じオブジェクトの位置情報 (座標) を特定する．連続するスナップショット画像中のオブジェクトの座標を取得することで，オブジェクトの移動を追跡する．ただし，連続するフレーム中でオブジェクトが登場しない (削除された) 画像までを 1 つの動作とする．

#### 3.3 学習者が実装したいオブジェクト操作の入力

学習者がイメージするオブジェクトの移動を，マウス移動座標を追跡することでオブジェクト操作の軌跡を取得する．現時点ではオブジェクト操作を入力するインターフェー

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	
マウスポインターに触れた	マウスポインターまでの距離
( )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	

スは実装できていないため、本研究の評価では既存の作品中のオブジェクトから、3.2.2 と同様に SIFT 特徴量を用いてオブジェクトが移動する軌跡を追跡し、オブジェクト操作の代替として使用する。

### 3.4 入力するオブジェクト移動を含む作品の抽出

学習者のオブジェクト操作の軌跡と、3.2.2 で計測したビジュアルプログラミング作品の画像中のオブジェクトの軌跡の類似度を DTW を用いて算出する。本論文では、類似度の高い作品と低い作品の違いを調査するため、閾値を定めない。

## 4. 実験

### 4.1 検索対象

本研究では、ビジュアルプログラミングサービスの中でも多くのプログラミング教育現場で利用されている Scratch を対象とする。また、Scratch のスナップショット収集を自動化するため、キー入力やマウス操作を必要としない作品のみを対象とする。具体的には、Scratch において動きブロック（例：指定した座標位置にオブジェクトを移動させるブロック、オブジェクトを回転させるブロックなど）を使用した作品を検索対象とし、表 1 に示すブロックを使用した作品を除外した。また、SIFT 特徴量におけるテンプレート画像として既知の画像を使用する作品のみを対象とするため、Scratch が提供するオブジェクトを使用した作品のみを検索対象とする。本研究における検索対象の作品は、Aivaloglou らの公開データセットに含まれる作品の中から、上述の条件を満たす 95,432 件の作品を用いる。[Aivaloglou\_2017]

### 4.2 結果：作品中のオブジェクトの動作間の類似度

オブジェクト操作データの時系列解析手法を評価として、3.3 でも述べた通り、既存の作品中のオブジェクトをオブジェクト操作データの代替として使用する。図 3 は、本研究で収集した 95,432 件の作品中、誌面の都合上ランダムにサンプリングした 65 件の作品に含まれる 118 のオブジェク

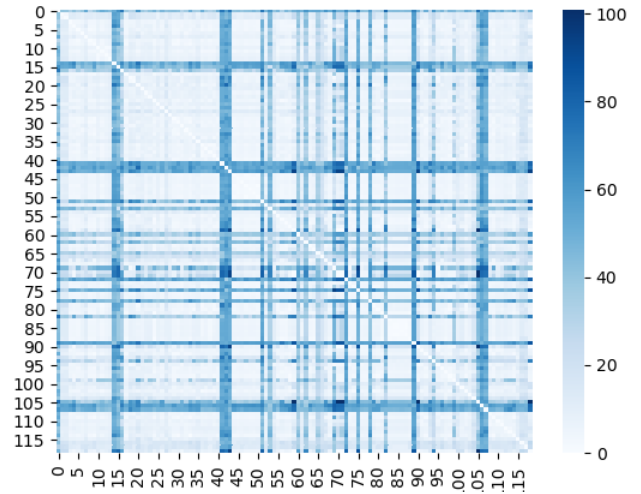


図3 動作の距離のヒートマップ

ト動作間の類似度を算出した結果をヒートマップで示す。縦軸と横軸は、それぞれ各動作を識別するための ID を示す。ヒートマップ中の色が薄いセルは類似度が高く、色が濃いほど類似度が高いことを示す。続く 4.2 節では、異なる作品で使用されたオブジェクト動作間の類似度の高いペアと、類似度が低いペアとして 4 つの事例を分析した結果を述べる。

**事例 1 - 類似度が高いオブジェクト動作の分析：**図 3 中のオブジェクト動作 ID が 33<sup>1</sup> と 114<sup>2</sup> の距離は 0.27 であり、類似度が高いオブジェクト動作のペアである。図 4 はそれぞれの X 座標の時系列変化と Y 座標の時系列変化を示す。図の横軸は時間（類似度の高い動作が出現するスナップショットの番号）を示し、縦軸は X 座標、Y 座標の値を示す。両方のオブジェクト動作は 3 つの命令処理から構成される。

1. 中央に位置するオブジェクトが下へ移動
2. 画面の端まで到達後、右上へ移動
3. 再度下へ移動

両プログラムはリミックス（再利用作品）元が同じであり、再利用元から一部変更は加えられているが、実装内容は完全に同じであり、類似度が高い同じ実装内容のオブジェクト動作を本手法で抽出できることを確認した。

**事例 2 - 類似度が高いオブジェクト動作の分析：**図 3 中のオブジェクト動作 ID が 24<sup>3</sup> と 81<sup>4</sup> の距離は 0.57 であり、類似度が高いオブジェクト動作のペアである。図 5 はそれぞれの X 座標の時系列変化と Y 座標の時系列変化を示す。横軸と縦軸は図 3 と同じである。両方のオブジェクト動作は 1 つの命令処理から構成される。

1. 左端に位置するオブジェクトが右へ移動

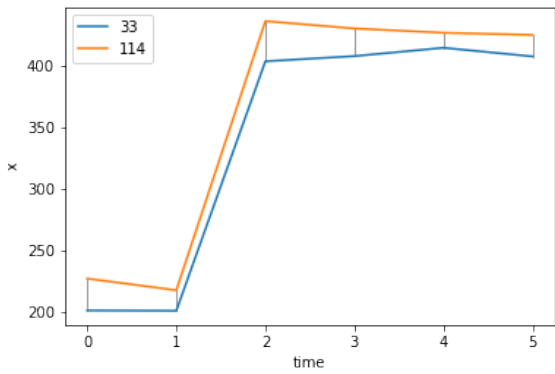
両プログラムのオブジェクト動作は類似しており、いずれもオブジェクトを移動する命令処理のブロックを使用している。ただし、ID.24 の作品は移動中に見た目を変更するブロックを用いている。このような場合、一般的なプログラム検索手法では無視するブロックを定義するなど、例外処理が必要となるが、本手法では同一の処理を目的とする異なる実装方法を収集可能であることを確認した。

<sup>1</sup> ID.33: <https://scratch.mit.edu/projects/99161197>

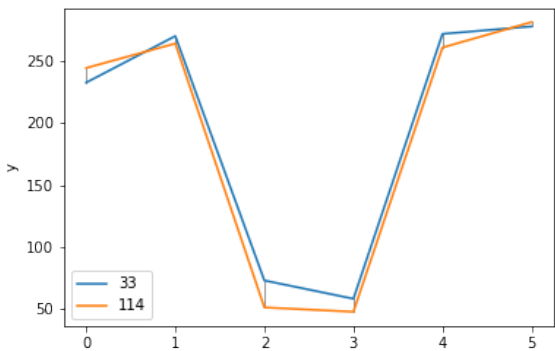
<sup>2</sup> ID.114: <https://scratch.mit.edu/projects/99232019>

<sup>3</sup> ID.24: <https://scratch.mit.edu/projects/99107978>

<sup>4</sup> ID.81: <https://scratch.mit.edu/projects/99143099>

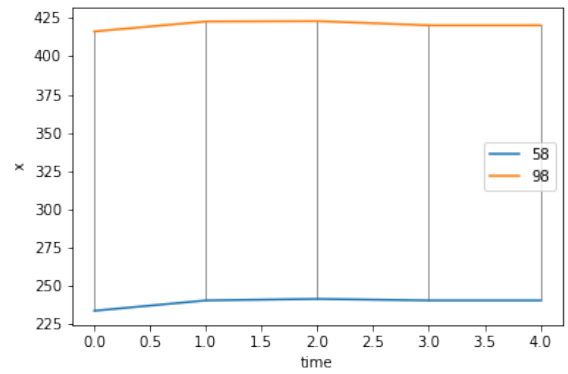


(a) X 軸方向の移動

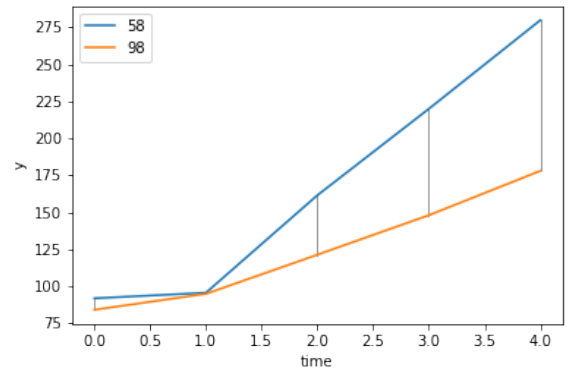


(b) Y 軸方向の移動

図 4 事例 1: オブジェクト動作 ID が 33 と 114 の時系列変化

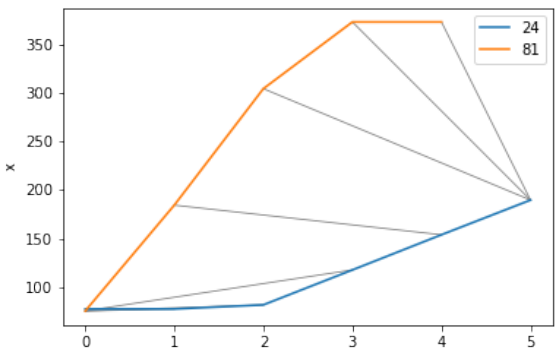


(a) X 軸方向の移動

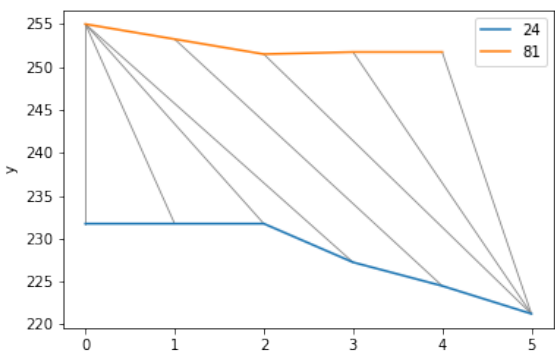


(b) Y 軸方向の移動

図 6 事例 3: オブジェクト動作 ID が 58 と 98 の時系列変化

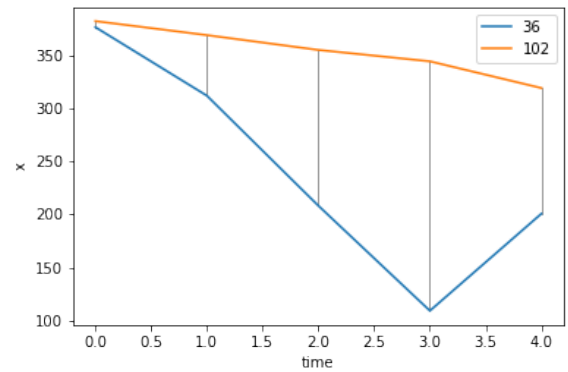


(a) X 軸方向の移動

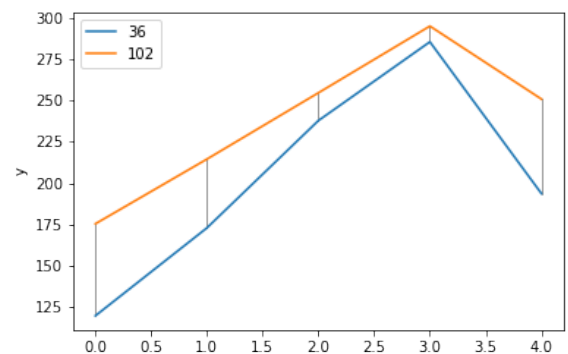


(b) Y 軸方向の移動

図 5 事例 2: オブジェクト動作 ID が 24 と 81 の時系列変化

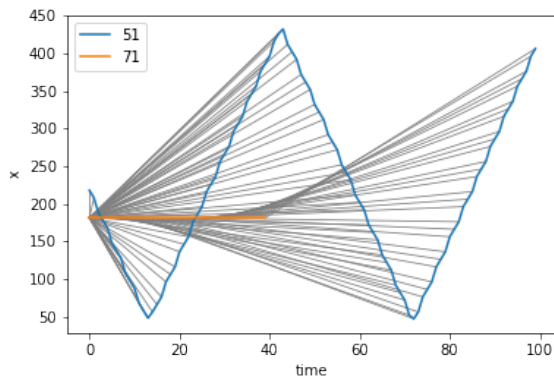


(a) X 軸方向の移動

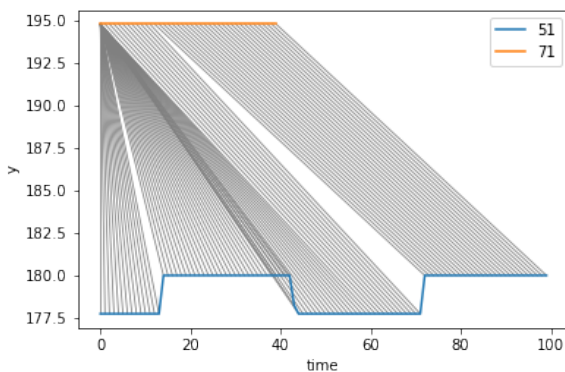


(b) Y 軸方向の移動

図 7 事例 4: オブジェクト動作 ID が 36 と 102 の時系列変化



(a) X 軸方向の移動



(b) Y 軸方向の移動

図 8 事例 5：オブジェクト動作 ID が 36 と 102 の時系列変化

**事例 3 - 類似度が高いオブジェクト動作の分析：**図 3 中のオブジェクト動作 ID が 58<sup>1</sup>と 98<sup>2</sup>の距離は 0.71 であり、類似度が高いオブジェクト動作のペアである。図 6 はそれぞれの X 座標の時系列変化と Y 座標の時系列変化を示す。横軸と縦軸は図 3 と同じである。両方のオブジェクト動作は 2 つの命令処理から構成される。

1. 上部に位置するオブジェクトが、右下へ小さい距離だけ移動
2. 下へ移動

両プログラムのオブジェクト動作は類似しており、いずれもオブジェクトを移動するブロックを使用している。しかし、移動のために使用しているブロック数が異なる。

**事例 4 - 類似度が高いオブジェクト動作の分析：**図 3 中のオブジェクト動作 ID が 36<sup>3</sup>と 102<sup>4</sup>の距離は 1.02 であり、類似度が高いオブジェクト動作のペアである。図 7 はそれぞれの X 座標の時系列変化と Y 座標の時系列変化を示す。横軸と縦軸は図 3 と同じである。両方のオブジェクト動作は 2 つの命令処理から構成される。

1. 上部に位置するオブジェクトが、左下へ移動
2. 折り返して右上へ移動

両プログラムのオブジェクト動作は類似しており、いずれもオブジェクトを移動するブロックを使用している。しか

し、ID.36 の作品はオブジェクトを回転するブロックを用いており、さらに移動先の座標を指定する際、乱数を用いており、多様な実装の作品を抽出できると考えられる。

**事例 5 - 類似度が低いオブジェクト動作の分析：**図 3 中のオブジェクト動作 ID が 51<sup>5</sup>と 71<sup>6</sup>の距離は 82.7 であり、類似度が低いオブジェクト動作のペアである。図 8 はそれぞれの X 座標の時系列変化と Y 座標の時系列変化を示す。横軸は時間（類似度の高い動作が出現するスナップショットの番号）を示し、縦軸は X 座標、Y 座標の値を示す。51 は常に座標の変化を続けているが、71 は座標が変化しないものである。

## 5. 考察

### 5.1 類似度の高いオブジェクトの特徴

4.2 節で述べた 5 つの事例において、類似度の高い動作は、いずれもスナップショット数が 5 枚~6 枚である一方、類似度の低い動作である事例 5 は、スナップショット数が 100 枚にわたる動作であった。スナップショット数が多いオブジェクト移動には複数の移動方向、移動距離が含まれ複雑であるため、類似度が低くなったと考えられる。本研究では、数ブロックで構成されるオブジェクト移動は検出可能であることを明らかにしたが、複雑なオブジェクト移動の検出は今後の課題である。

### 5.2 制約

本研究では、学習者のキー入力やマウス操作を必要とする作品の動作を捉えることはできない。また、4.2 で挙げた動作 36 は、乱数を生成するブロックを含んでいたが、そのような動作は実行の度に結果が変化するため、検索対象外の作品とする必要がある。

## 6. おわりに

本論文では、ビジュアルプログラミングの学習における直感的なプログラム作品検索システム実現に向けて、時間経過によるオブジェクトの座標変化を記録した時系列データを利用する検索手法を提案した。実験では、既存の作品中のオブジェクトを用いてオブジェクトが移動する軌跡を追跡し、DT をも用いてサービス中で公開される作品の中からオブジェクトの移動が類似する作品を分析した。その結果、類似度の高いオブジェクト動作の場合、オブジェクトの動作は同じでも異なるブロックを使用した実装の作品を収集できることを確認し、本手法の有効性を確認した類似と判断できるオブジェクト移動の閾値を実験的に調査し、複雑な動作の検索を実現する時系列データの解析手法を検討する。

<sup>1</sup> ID.58: <https://scratch.mit.edu/projects/99123462>

<sup>2</sup> ID.98: <https://scratch.mit.edu/projects/99124029>

<sup>3</sup> ID.36: <https://scratch.mit.edu/projects/99145792>

<sup>4</sup> ID.102: <https://scratch.mit.edu/projects/99120074>

<sup>5</sup> ID.51: <https://scratch.mit.edu/projects/99141356>

<sup>6</sup> ID.71: <https://scratch.mit.edu/projects/99120562>

## 参考文献

- [1] S. E. Sim, M. Umarji, S. Ratanotayanon, and C. V. Lopes, “How Well Do Search Engines Support Code Retrieval on the Web?” *ACM Transactions on Software Engineering and Methodology*, Vol.21, No.1, pp.1-25, 2011.
- [2] 杉浦学, 松澤 昭, 岡田健, 大岩元ほか: アルゴリズム構築能力育成の導入教育: 実作業による概念理解に基づくアルゴリズム構築体験とその効果, *情報処理学会論文誌*, Vol. 49, No. 10, pp. 3409–3427 (2008).
- [3] Yang, S., Domeniconi, C., Revelle, M., Sweeney, M., Gelman, B. U., Beckley, C. and Johri, A.: Uncovering Trajectories of Informal Learning in Large Online Communities of Creators, *In Proceedings of the 2nd Conference on Learning @ Scale (L@S'15)*, pp. 131–140 (2015).
- [4] Aivaloglou, E., Hermans, F., Moreno-Le on, J. and Robles, G.: A Dataset of Scratch Programs: Scraped, Shaped and Scored, *Proceedings of the 14th International Conference on Mining Software Repositories (MSR'17)*, pp. 511–514 (2017).
- [5] Robles, G., Moreno-Le on, J., Aivaloglou, E. and Hermans, F.: Software clones in scratch projects: on the presence of copy-and-paste in computational thinking learning, *Proceedings of the 11th International Workshop on Software Clones (IWSC'17)*, pp. 1–7 (2017).
- [6] Dasgupta, S., Hale, W., Monroy-Hern andez, A. and Hill, B. M.: Remixing As a Pathway to Computational Thinking, *Proceedings of the 19th Conference on Computer-Supported Cooperative Work & Social Computing (CSCW'16)*, pp. 1438–1449 (2016).
- [7] Troiano, G., Snodgrass, S., Argimak, E., Robles, G., Smith, G., Cassidy, M., Tucker-Raymond, E., Puttick, G. and Hartevel, C.: Is My Game OK Dr. Scratch?: Exploring Programming and Computational Thinking Development via Metrics in Student-Designed Serious Games for STEM, *Proceedings of the 18th International Conference on Interaction Design and Children (IDC'19)*, pp. 208–219 (2019).
- [8] Troiano, G., Chen, Q., Vargas-Alba, A., Robles, G., Smith, G., Cassidy, M., Tucker-Raymond, E., Puttick, G. and Hartevel, C.: Exploring How Game Genre in Student-Designed Games Influences Computational Thinking Development, *Proceedings of the Conference on Human Factors in Computing Systems (CHI'20)*, pp. 1–17 (2020).
- [9] 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).
- [10] David G. Lowe.: “Object Recognition from Local Scale-Invariant Features,” *Proc. of the International Conference on Computer Vision*, pp.1-8, 1999.