

振り返りと体験共有のための ビジュアルプログラミング画面要約手法

高田 秀志^{1,a)} 小林 冠太¹ 市村 真希¹

概要: 小学校でプログラミング教育が導入されたことなどに伴って、Scratch などのビジュアルプログラミング環境を利用したプログラミングワークショップに多くの子ども達が参加するようになった。一方で、子ども達のプログラミング体験はその場限りになってしまうことが多く、ワークショップへの参加をきっかけに自発的にプログラミングに取り組むようになることはそれほど多くない。また、プログラミングワークショップに参加した子ども達と、参加したことのない子ども達の間で体験の共有が十分に行えていないため、ワークショップに参加しようという動機づけにつながらず、十分に裾野が広がっているとは言い難い。このような課題を解決するために、本研究ではワークショップでどのような体験をしたのかを動画としてまとめ、ワークショップ参加後の振り返りや子ども達の間での体験共有につなげる方策を提案する。本稿では、このような体験動画においてプログラミングを行っている PC 画面に着目し、ワークショップ中に獲得された画面キャプチャから数分程度に要約された動画を生成する手法と、その評価について述べる。

キーワード: プログラミング学習, 画面キャプチャ, 動画要約, 振り返り

A Summarizing Method for Visual Programming Screen to Support Reflection and Experience Sharing

HIDEYUKI TAKADA^{1,a)} KANTA KOBAYASHI¹ MAKI ICHIMURA¹

Abstract: With the introduction of the programming education to primary schools, many students are now participating in a programming workshop utilizing visual programming environments such as Scratch. On the other hand, their programming experience is often limited in the period of a workshop, and there are not so many cases where they keep working on their own projects spontaneously after attending such a workshop. In addition, many students are not motivated but still hesitating to attend the workshop because the experience of students who have attended the workshop is not effectively shared with those who have not. In order to overcome this situation, we propose to take measures of sharing the experiences among students and supporting reflection on their activity by using a movie highlighting impressive scenes of what was experienced during a workshop. Focusing on the PC screen on which students are working on programming, this paper presents a method to generate a few-minutes long movie extracted by summarizing a captured screen taken during a workshop, along with preliminary evaluation results of its usefulness.

Keywords: Programming learning, Screen capture, Reflection

1. はじめに

社会状況の変化や技術の進展に伴い、コミュニケーション

ン力、問題発見・解決力、協働力などの「21世紀型スキル」と呼ばれる能力を学校教育においていかに育成するかが課題となっている。これに伴い、日本においても、「主体的・対話的で深い学び」(文部科学省)が提唱され、アクティブラーニングやプログラミング教育など、「学習における体験」を重視した教育手法・学習方法が注目されている。特

¹ 立命館大学情報理工学部
Ritsumeikan University, Kusatsu, Shiga 525-8577, Japan
^{a)} htakada@cs.ritsumeai.ac.jp

に、プログラミングは創造的思考力 [1] を育成する方法として注目されており、Scratch[2] などのビジュアルプログラミング環境を活用したワークショップなどが広く開催されている。

我々は、過去 15 年以上に渡って、主に小学生を対象としたプログラミングワークショップの開催や学校におけるプログラミング授業の実施を通じてプログラミング学習の実践を行っている。このような取り組みを通して、学習活動への児童らの満足度は高く、知識獲得型の学習とは違った能力を育成することに貢献できていると考えているものの、自らプログラミングに継続して取り組めるようになる持続性の維持や、プログラミングへの興味をより多くの子ども達に広げていくことが課題になっている。特に、「クリエイティブ・ラーニング・スパイラル」[3] において強調されているように、持続的な創造活動につなげていくためには、振り返りの活動が重要である。

これを踏まえて、本研究では、児童の振り返り活動を支援するために、ワークショップ中の体験を数分にまとめた動画を生成し、活用することを考える。動画には、PC 上でプログラミングを行っている画面に加えて、カメラやマイクで収録した喜びや驚きの様子なども含めることを想定する。本稿においては、特に Scratch を利用している PC 画面を要約することに着目し、ワークショップ中に獲得された数時間のスクリーンキャプチャから数分程度に要約された動画を生成する手法を提案する。また、本手法を実際のワークショップで獲得された PC 画面に適用し、どの程度有用な動画に要約できるのかを評価した結果について述べる。

2. プログラミング学習における振り返りと体験共有

2.1 問題意識

現在、我々が NPO 法人と連携して実施している小学生を対象としたプログラミングワークショップにおいては、ブロックを組み合わせることで視覚的にプログラムを制作することのできる Scratch を用いている。ワークショップでは、構築主義 [4]、および、その考え方を基盤とした「発想」「創作」「遊び」「共有」「振り返り」のプロセスで構成される「クリエイティブ・ラーニング・スパイラル」の枠組みに基づき、例えば「オリジナルゲームを作ろう」というテーマに基づいて、プログラミングに必要な基本概念（順次実行、繰り返し、条件分岐）を使いながら、ゲームとして一貫するような作品を参加者自身のアイデアを取り入れながら構築（「発想」「創作」「遊び」）し、最後に発表会（「共有」「振り返り」）を行っている。このような活動において、上述した学びのプロセスが実現されているものの、創造的思考力を向上させるには、自宅等でもワークショップの体験を振り返り、継続的にプログラミングに取り組む

など、スパイラルを自ら回していけるようにすることが必要である。

また、プログラミングワークショップに参加するに当たって、「自分の子どもにプログラミングが理解できるのか」という問い合わせを父母から受けることが多い。「何かを作ることによって自発的に学ぶ」という構築主義の考え方によれば、プログラミング自体に対する理解よりも、プログラミングという過程を通して、様々な知識を自ら獲得できるような体験をすることが重要であり、我々のワークショップもそのような意図を持って構成しているが、実際に参加してみないとその意義が理解されにくい。ワークショップに参加した子どもたちの目線で、どのような体験が行われたのかをこれから参加しようとしている人たちにも伝わるような形で共有できるようにする必要がある。

2.2 関連研究

Scratch 上での創造的な活動を振り返る機会を子ども達に与えるためのシステムとして、Scratch Memories が開発されている [5]。Scratch Memories では、初めて共有したプロジェクト、初めてお気に入り登録されたプロジェクト、最も多くのスクリプトを持つプロジェクトなど、Scratch の Web サイト上でのユーザの体験に焦点を当てた動画を個別に生成することができる。これにより、子ども達の振り返り活動を促進するだけでなく、Scratch コミュニティへ参加し続けることへの動機づけを与えることを狙っている。

このように、Scratch Memories では長期間に渡る子ども達の体験を振り返ることを可能にしているが、「今日の」あるいは「最近の」ワークショップへの参加などの体験を振り返ることを可能にすることも重要である。本稿で述べる手法により、ワークショップが終了した直後に、振り返り活動を行うことを可能にするような動画を生成することを目指す。

また、我々は以前、プログラミングによる作品の制作過程をワークショップの参加者間で共有できるようにすることにより、「共有」と「振り返り」を促進するような教室内 SNS を構築した [6]。これは、ワークショップ中に作成した作品の写真や動画、コメントなどをタブレット端末で投稿・共有できるようにしたものであり、自宅に戻ってからも保護者や他の参加者とワークショップ中の体験を共有できるようになることを意図した。

このシステムを利用することにより、ワークショップが行われた日に振り返りを行うことも可能になるが、作品の写真や動画を自分で投稿しなければならぬため、作品作りに集中することを阻害する要因になり得る。これに対して、本研究では、自動的に獲得された PC 画面から振り返りに有用な場面を自動的に抽出することを可能にすることにより、ワークショップ中に子ども達が作品作りに集中することができるようにする。

3. プログラミング画面の要約手法

3.1 手法の全体像

本節では、プログラミングを行っている画面の録画から、プログラミング体験を表現するような動画ファイルを生成する手法を述べる。画面の録画には、Windows あるいは MacOS に搭載されているキャプチャソフトウェアをそのまま利用することを想定する。キャプチャされた画面録画は、ワークショップの長さに応じて数時間の長さになる可能性があるが、そこから、振り返り活動に有用な場面を抽出し、数分間の動画ファイルを生成することを考える。

通常、ワークショップにおいては、Scratch を開いたほぼ白紙の状態から、スプライトやスクリプトの追加、背景の追加や変更などを行いながら機能を追加していくことにより、作品を作成していく。したがって、このような作品の制作過程や、プログラムを実行したときの画面などを振り返りに有用な場面として抽出する。

画面録画からの有用な場面の抽出には、画像中の特徴点の抽出や、画像間の特徴点マッチングを行うことができる OpenCV を用いる。このような機能を用いることにより、画面上で Scratch のスプライトやプログラミングブロックがどのように動いたのかを定量化することができるようになる。

本稿で提案するプログラミング画面の要約手法は、以下の2つのステップで構成される。

- 画面録画全体から、連続する画面画像内の特徴点の動きを検出することにより、要約動画に含まれる可能性のある**候補場面**を抽出する。
- 抽出された候補場面から、候補場面間で特徴点を比較し、似た場面を除外することにより、要約動画に含まれるべき**重要場面**を抽出する。

3.2 候補場面の抽出

本手法の第1ステップでは、要約動画に含まれる可能性のある候補場面を抽出する。動画ファイルは、一般に、フレームと呼ばれる静止画の連続として捉えることができる。したがって、候補場面の抽出は、振り返り活動に意味のあると考えられる連続したフレームを決定する問題として表現することができる。

候補場面を抽出するために、Scratch 画面に表示されているスプライトとスクリプトブロックの動きに着目する。スプライトやスクリプトブロックは、OpenCV によって特徴点として認識することが可能である。また、それらの動きは、隣り合うフレーム同士で特徴点の座標を比較することにより検出することができる。プログラミング体験の振り返りという意味においては、スクリプトがどのように構築されていったのか、また、そのスクリプトがどのように実行されたのか（ステージ上でスプライトがどのように動

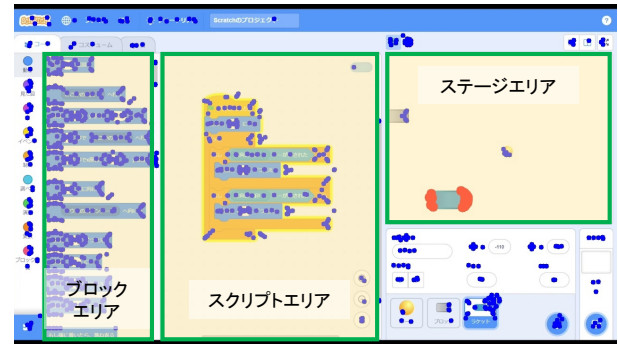


図 1 Scratch のウィンドウと検出された特徴点

いたのか) が要約動画に含まれるべきであると考えられる。これに基づき、図 1 に示すように、Scratch のウィンドウの中身を「ブロックエリア」「スクリプトエリア」「ステージエリア」の3つの領域に分けた上で、最終的な要約動画に含まれるべき場面であるかどうかの判定を行う。

通常、PC 上で画面キャプチャにより生成される動画ファイルは 60 フレーム/秒であるが、それぞれのフレームを処理すると非常に時間がかかることが予想される。したがって、本手法では、全体の画面録画から 1 秒間隔でフレームを抜き出して処理を行うこととする。候補場面の抽出においては、それぞれのフレームから特徴点を抽出し、隣り合うフレーム同士でそれらのマッチングを行うことにより、特徴点の移動距離を求める。図 1 において、青い点は検出された特徴点を示し、赤い点は前のフレーム (1 秒前) から 10 ピクセル以上移動したと判定された特徴点を示している。もし、10 ピクセル以上移動した特徴点が以下の条件を満たした場合、抽出する候補のフレームとする。

- そのような特徴点がスクリプトエリアに 10 個以上あり、かつ、ブロックエリアおよびステージエリアにおける数より 5 倍以上である場合
- そのような特徴点がステージエリアに 5 個以上ある場合

1 番目の条件はユーザがマウス操作によってプログラミングブロックを移動させ、スクリプトを構築している段階であることを示している。また、2 番目の条件は、スクリプトの実行によってステージ上のスプライトが動作している状態を示している。

その後、もし候補のフレームが 4 個以上 (4 秒以上) 連続している場合に、それらを候補場面として抽出する。

なお、これらの数値は経験的に決定したものであり、画面上の動きの多さに応じて個別に調整することも考えうる。

3.3 重要場面の抽出

第1ステップで抽出した候補場面は、単にスプライトやスクリプトブロックの動きに着目しているため、それらをつなぎ合わせて一つの動画にすると、似たような場面が長く連続してしまう冗長なものになる可能性がある。そこ

で、第2ステップでは、隣り合う候補場面同士を比較し、もし同じ特徴点が多く含まれる場合は、そのような場面を削除するようにする。

候補場面同士の比較は、候補場面に含まれるフレームのうち、真ん中に位置するものの特徴点を比較することにより行う。例えば、候補場面が240フレーム(4秒間)で構成されている場合、120フレーム目の静止画を比較に用いる。もし、隣り合う候補場面でマッチングされた特徴点の数が一定の閾値を上回ったとき、それらの場面は似ているとして、後ろ側の候補場面を削除する。これを繰り返し行い、残った候補場面を重要場面として抽出する。また、重要場面をつなぎ合わせたものを生成する要約動画とする。

場面を削除する際の判定に用いる閾値は、最終的な要約動画の長さに影響を与える。もし、閾値が小さくなれば、削除される候補場面の数が多くなるため、要約動画は短くなる。

3.4 実装環境

本手法は、OpenCV ライブラリを Python 上で利用することで実装した。特徴点抽出の手法としては、AKAZE (Accelerated KAZE) を用いている。また、特徴点マッチングには総当たり方式である BF (Brute-Force) Matcher を用いている。

実装したプログラムを、Intel 社の第8世代 Core i7 を搭載した MacBook Pro で実行したところ、約2時間半の画面録画から約5分の要約動画を生成するのに、約35分の時間を要した。ただし、OpenCV による特徴点抽出や特徴点マッチングはマルチスレッド化されていないため、マルチコア CPU によって複数の画面録画に対する要約動画の生成を同時に行うことで、一つの要約動画を生成するのに必要な平均時間を削減することが可能である。

4. 評価

4.1 評価方法

本節では、2021年11月7日および2021年11月21日に行われたプログラミングワークショップで実際にキャプチャされた画面録画を用いて、本手法により生成された要約動画がどの程度適切に構成されているかを評価した結果について述べる。

ワークショップにおいて、参加者は設定されたテーマに従って、講師の説明を聞きながら順を追って機能を追加していくという形態で作品作りを行った。テーマは、「からくり時計」(図2)と「ブロック崩し」(図3)であり、前者に対して9、後者に対して10の画面動画が収集された。それぞれの画面録画の長さはおよそ2時間半である。似た候補場面を削除するのに用いるマッチング特徴点数の閾値は、要約動画の長さが3分から5分程度になるように、それぞれの画面録画毎に値を変えて設定した。



図2 「からくり時計」の画面例

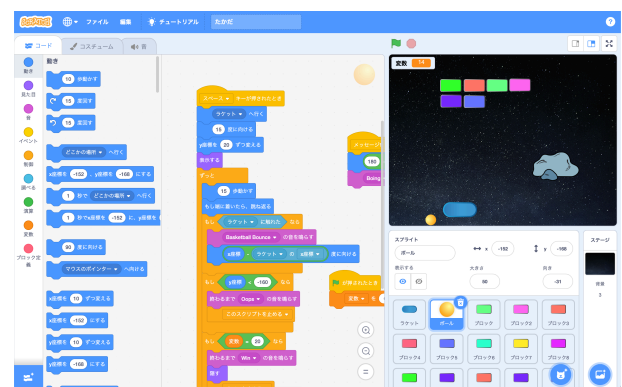


図3 「ブロック崩し」の画面例

評価は2つの観点で行った。一つ目の観点は、要約動画の中に含まれることが期待される場面(期待場面)が実際にどの程度含まれているかを適合率および再現率で客観評価したものである。適合率および再現率を求めるための計算式を以下に示す。

$$\text{適合率} = \frac{\text{要約動画に含まれていた期待場面数}}{\text{要約動画に含まれていた全場面数}}$$

$$\text{再現率} = \frac{\text{要約動画に含まれていた期待場面数}}{\text{画面録画に含まれていた期待場面数}}$$

もう一つの観点は、ワークショップの参加者に対して、生成された要約動画が振り返りにどの程度有用であったのかをアンケートで尋ねた主観評価である。アンケートは Google Forms により2022年1月に実施し、参加者は要約動画を視聴した後、アンケートへの回答を行った。

4.2 評価結果

客観評価においては、最終的な要約動画に含まれることが期待される場面として、「からくり時計」に対して8場面(表1)、「ブロック崩し」に対して9場面(表2)を設定した。これらの場面は、講師が作品制作の過程で順を追って説明していった内容に即している。これらの場面对して、それぞれの場面が要約動画に含まれているか否かによって、適合率および再現率を求めた。その結果を図4に示す。図中に示すそれぞれの点は、19の要約動画に対する適合率および再現率を示している。すべての要約動画

に対する適合率の平均は 0.303、再現率の平均は 0.647 であった。

表 1 「からくり時計」で抽出すべき場面
期待される場面の内容

長針が回るようにする
短針が回るようにする
数字を配置する
定刻になるとスプライトが飛び出てくるようにする
定刻になると音が鳴るようにする
ある時間になると背景が変わるようにする
ある時間になると動き出すスプライトを追加する
完成した作品を実行する

表 2 「ブロック崩し」で抽出すべき場面
期待される場面の内容

左右の矢印キーでラケットが動くようにする
ボールが動き回るようにする
ラケットに触れるとボールが跳ね返るようにする
ラケットに触れると正しくボールが跳ね返るようにする
ボールが落ちるとゲームオーバーにする
ラケットにボールが当たるときにゲームオーバーになったときに音を鳴らす
ブロックにボールが当たると消えるようにする
ブロックに触れるとボールが跳ね返るようにする
完成した作品を実行する

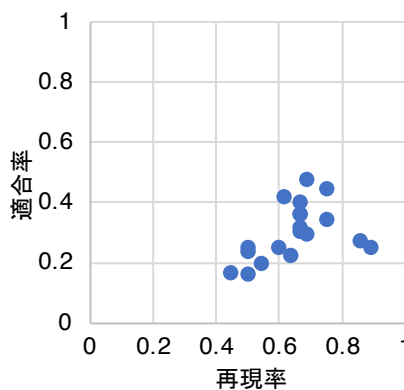


図 4 適合率・再現率による客観評価の結果

この結果により、期待された場面の約 65%は抽出することができている一方で、抽出された場面のうち約 70%は期待された場面に合致していないと言える。これは、子ども達がプログラミングの活動を振り返る上で有用な場面は抽出されているが、動画の中には冗長な場面が多く含まれていることを意味する。ただし、同じ期待場面に対して複数の場面が抽出されていた場合には、1つの場面のみを正しく抽出されたものとみなしていたため、適合率が下がる要因にはなっているが、これらは全く意味がない場面というわけではない。実際的な要約動画の中には、一

つの期待画面が複数の重要場面にまたがって抽出されており、同じような場面が長く続くような場合が含まれていたため、要約手法における第 2 ステップにおいて、似たような場面を除外する方法に改良の余地があると考えられる。また、Scratch 上でスプライトや音を選択している画面が抽出されている場合もあったが、そのような場面は期待場面の中には入っていないものの、体験を振り返るという点では意味のある場面と考えることもできる。今後、要約動画の適切性についてどのような指標を用いるかを再検討することが必要である。

次に、主観評価におけるアンケートの質問を表 3、回答結果を図 5 に示す。ワークショップの参加者数は 15 人であり、6 人からアンケートへの回答が得られた。回答結果からは、多くの参加者が活動の振り返りに対して肯定的な回答をしていることが分かる。自由記述による感想としても、「楽しく体験できたと思います」「ちゃんと作れたかを思い出して振り返れました」「もうちょっと改良したい」との記載があった。一方で、一部の参加者にとっては、作品を改良してみるなど、次のステップには必ずしもつながっていない様子も見られた。また、感想として「あまりできなかったことを思い出してしょんぼりしました」という記載もあり、振り返りを行うことが逆効果になってしまわないように、単にできた・できなかったではなく、ワークショップへの参加目的に対する理解を深めるような取り組みも必要であると考えられる。

表 3 アンケートの質問

観点	質問
学んだこと	プログラミングワークショップで学んだ作り方を思い出すことができましたか？
楽しさや嬉しさ	プログラミングワークショップでの楽しさや嬉しさを思い出すことができましたか？
改良点	もっとこうしたら面白い作品ができたなど思い付くことができましたか？
次の作品	次に作ってみたい作品が思い付きましたか？
体験動画	今後のワークショップでも自分の体験動画を見たいですか？

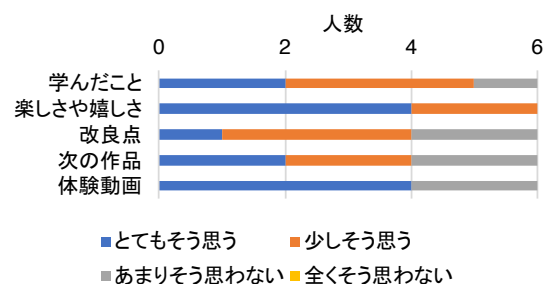


図 5 アンケートへの回答結果

5. おわりに

本稿では、Scratch を用いたプログラミング活動において、子ども達のワークショップでの体験を振り返ることを支援するためのプログラミング画面の要約手法について述べた。生成された要約動画を評価したところ、OpenCV を用いた特徴点抽出により振り返りに重要な場面は抽出されているものの、冗長な部分が多いことが分かった。

要約動画の精度をより高めていくには、候補場面抽出、および、重要場面抽出において手法の改良が必要である。また、今回用いたルールベースの方法ではなく、機械学習を用いて要約動画に含めるべき場면을抽出する方法についても検討を行う予定である。

謝辞 本研究は JSPS 科研費 21K12165 の助成を受けたものです。また、ワークショップ中の PC 画面の録画や要約動画の視聴、アンケートへの回答に協力して下さった参加者および保護者の皆様に感謝いたします。

参考文献

- [1] Resnick, M. and Robinson, K.: *Lifelong Kindergarten: Cultivating Creativity through Projects, Passion, Peers, and Play*, MIT Press (2017).
- [2] Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B. et al.: Scratch: programming for all, *Communications of the ACM*, Vol. 52, No. 11, pp. 60–67 (2009).
- [3] Resnick, M.: All I Really Need to Know (About Creative Thinking) I Learned (By Studying How Children Learn) in Kindergarten, *Proceedings of the 6th ACM SIGCHI Conference on Creativity & Cognition*, pp. 1–6 (2007).
- [4] Papert, S. A.: *Mindstorms: Children, computers, and powerful ideas*, Basic books (2020).
- [5] Dhariwal, S.: Scratch Memories: a Visualization Tool for Children to Celebrate and Reflect on Their Creative Trajectories, *Proceedings of the 17th ACM Conference on Interaction Design and Children*, pp. 449–455 (2018).
- [6] Tsutsui, K. and Takada, H.: A classroom SNS to promote reflective activity in programming learning for children, *Research and Practice in Technology Enhanced Learning*, Vol. 13, No. 1, pp. 1–18 (2018).