

能動的学習を支援するための アルゴリズムアニメーションの拡張法に関する考察

井上 勝行 魚井 宏高 首藤 勝

大阪大学 大学院基礎工学研究科

これまでにさまざまなアルゴリズムアニメーションシステムが研究・開発されているが、それらを実際に利用している例は現在でも多くはなく大きな効果があったことが報告されたことも少ない。しかしアルゴリズムアニメーションは、まだまだ未成熟な技術であり実用的な目的を持って開発されたシステムがほとんどないことを考えれば、これは半ば当然のことである。アルゴリズムアニメーションは直観性が高く観察者を惹き付ける効果も大きいことから、例えば学習環境に適応したシステムを用いて実際に即した状態での効果を評価することにより異なる結果が出るということが十分に予想される。このような考えを基に本稿では、アルゴリズムアニメーションの学習環境への効果的な採り入れ方について考察する。その結果アルゴリズムアニメーションには、内容に教科書との一貫性を持たせて対応関係を明確にすること、複数の view を用いる際にはその組み合わせに適した表現を用いること、知識の一方的な提示だけではなく学習者がシステムと対話しながら情報を検索できるようにすることなどが求められるとの知見が得られた。

Enhancing Algorithm Animation to Support Active Style Learning of Computer Algorithms

KATSUYUKI INOUE, HIROTAKA UOI and MASARU SUDO

Graduate School of Engineering Science, Osaka University

Various algorithm animation systems have been developed. They, however, are seldom used in practice, and few cases are reported that prove their noteworthy effectiveness. But it is no wonder because algorithm animation is immature technique and few systems have been developed for practical use. Algorithm animation is featured with high intuitiveness and strong attractiveness, so we expect that it will be more effective to use practical system for learning algorithms, for example. In this paper, we discuss methodology of enhancing algorithm animation to take it into algorithm learning environment. Our conclusion consists of three principles. Firstly, the practical algorithm animation systems for learning should hold consistency between animation and textbook in their contents and representations and should clarify their correspondency. Secondly, displaying more than one views simultaneously, the presentation should be optimized to the combination. And thirdly, the systems should support not only presentations of the knowledge but also some interactive method for inquiring information.

1. はじめに

フィルム「Sorting Out Sorting」²⁾以来、アルゴリズムの本質をいかに視覚的に表現するかと

いうことを研究の焦点に据えて、多くのアルゴリズムアニメーション(以下 AA)システムが研究、開発されてきた。例えば、Path Transition という簡潔で表現力の豊かなパラダイムを備えた

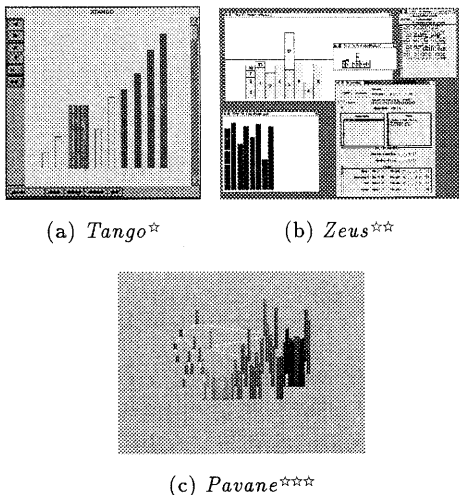


図1 代表的な AA システムの画面
Fig. 1 sample screens of famous AA systems

Tango¹⁰⁾(図1-(a))や、Interest Event ごとに対応する動画上の変化を定義することによりほとんどあらゆる視覚的表現を実現可能な Zeus³⁾(1-(b))などは一般的なアプローチによるシステムであるといえる。独特なアプローチによる研究としては、Hyper Card 上に手作りのアニメーションを表現することにより対話性を始めとするさまざまなユーザー親和性の向上を狙った AACB⁵⁾や、宣言的な規則を用いた純粋に数学的な方法により計算を可視化することに適する Pavane⁸⁾(図1-(c))、またプログラムから視覚的表現へ変換するだけでなく、その逆変換も可能なパラダイムを提案する TRIP2a^{12),13)}などが挙げられる。そのほかアニメーションの設計や実装を容易にする研究も盛んであり、単純な言語により操作可能なアニメーション用ツールキットを用いて手軽にアニメーションを作成できる Goofy⁴⁾や、カプセル化や多重定義、そしてそれらの実体化や継承による再利用といったオブジェクト指向の考え方によって視覚的表現の構築を容易にすることを狙った GAIGS⁷⁾などが開発されている。このようにさまざまなアプローチによるさまざまなシステムが研究されてきたが、それらのシステムはおもにアニメーションの作成にかかる手間と時間からアルゴリズムの教育支援を主用途として挙げるが多かった。

ところが Badre らの調査¹⁾によれば、実際に基本的なアルゴリズムの講義を担当している教官の多くは AA を教材として使用していない。これは従来、AA を経験的あるいは形式的に評価した研究が少なかったことも原因になっていると思われるが、近年になって行なわれたいくつかの実験においては AA の教育的効果を否定しかねない結果が提出されていることも事実である。Stasko, Lawrence らは 6), 11) において、アニメーションを使用した場合と使用しない場合における被験者の試験の成績を調査しているが、初学者に対しては効果が少なかったことや、スライドに対して際だった優位性が見られないことなどを報告している。しかしこのことをもって AA に教育的効果が少ないと直ちに断定することはできない。なぜならば上記のような実験に使われたほとんどのシステムは、アニメーション技術の研究を目的として開発されたもので、応用の場面を具体的に想定した上では開発されていないのである。したがって AA を教材として評価するためには、対話性や表現力、伝達力を向上させるなど用途に合わせて設計したシステムを用いて実験する必要がある。もちろん AA システムの中には、AACB や GAIGS, Goofy などのように教育現場で実用することを目的として開発されたものもある。AACB はアルゴリズムの教科書に示された図の対話型版を提供することを、GAIGS は学生が作成した抽象データ型を用いたプログラムの動作を視覚的にモニタリングすることを、また Goofy は AA の作成を演習課題とするために AA 作成の容易化することをそれぞれ目的としており、それぞれの著者はある程度経験的な評価においてよい結果が出たと主張している^{4),5),7)}。しかし AA は本来知識の一表現法としてだけでなく、ソフトウェアの動的な振る舞いに基づいて動画を生成する技術としても重要で

★ <http://www.cc.gatech.edu/gvu/softviz/algoanim/bsort.gif>

★★ <http://www.research.digital.com/SRC/zeus/bin-packobl.twofifth.gif>

★★★ <http://swarm.cs.wustl.edu/vislab/gallery/pics/quicksort2.Gif>

ある。アルゴリズムに与える初期入力の変更や対話的な実行制御、情報の検索とその視覚的な表示など、学習者が能動的に実験するための学習教材としての価値が重要であると思われる。そこで本稿では、AAの教材としての価値を実験に基づいて評価するためにAAがどのように設計され、システムとしてどのような機能を備えるべきなのかについて考察する。

2. 学習支援のための対話型システムとしてのAAシステム

AAは基本的にアルゴリズムの実装の動作に基づいて動画を生成する技術である。観察者の理解を促すという同じ目的をもつため、その内容の設計も説明図と同様の観点から行なわれる。したがって厳密な知識を伝達したり複数の媒体や表現を効果的に併用したりするためには、AAに何らかの拡張を加える必要がある。本節では、AAシステムをより良いアルゴリズム学習教材にするための拡張法について考察する。

2.1 学習環境を実現するためのAAの拡張法

AAは実際の動作に基づいていることから、単に連続した説明図をスムーズアニメーション化した動画とは、

- 問題のインスタンスの解かれ方を表現対象とすること
- 多様な入力に対応すること

などの点で特徴づけられる。つまり本来にAAは、概念的な説明よりもさまざまな状況において発生する現象を提示する手段であるといえる。したがって観察者がアルゴリズムを学習する際には、AAは一種のシミュレータとして機能することにより観察者の経験的な知識獲得を支援することができる。しかし従来のAAシステムには、観察者が獲得した経験的な知識を体系化することを支援するという配慮がほとんどなされていなかった。このことが従来のAAを評価しようとする実験結果に与えた影響が大きいと思われる。つまりAAやAAを提供するシステムが閉じていたために、観察者が時おり現れる重要な場面を漫然と見過ごし

てしまったり、たとえ気づいたとしても観察者自身が書籍などのAA以外の媒体からその意味を調べなければならなかった。これは発見的学習のマイナス面が現れやすい状況であったと考えられるのである。もっとも、こういった問題に対処し得る可能性がある機能もいくつか考案されている。例えばBrownのZeusシステムでは説明文を添付する機能や、MVCモデルに基づいた設計により、

Multiple Views (以下MV)

アルゴリズムの異なった側面を複数のウィンドウに同期させながら表示する機能

Multiple Algorithms (以下MA)

複数のアルゴリズムの振る舞いなどを同期させて表示する機能

Multi-View Editing

画面上のオブジェクトを直接的に操作することにより状況を強制的に変化させる機能などが実装されている³⁾。これらの機能の効果を示すデータは筆者には見つけることができなかったが、機能を適切に強化し内容を慎重に設計することによって大きな効果をもたらすことができると考えられる。

では次に、アルゴリズム学習におけるAAの利用形態について考察する。もっとも単純な形態としては、講義における資料として口頭で説明しながら学習者に示すことが考えられるが、前述したようにスライドやビデオなどの媒体に比べて特別な効果を期待することは難しい。そこでここでは演習などの能動的学習形態においてAAシステムを利用することを考えることにする。つまり学習者に実際にシステムと対話させることによって、さまざまな入力に対してアルゴリズムがどのように振る舞うかを学習者自身に確認させるという形態である。この場合学習者は、分岐を引き起こす条件やループ不変式、あるいは計算そのものの意味を視覚的に確認することができる。このような形態における評価実験もすでに行なわれている^{8),11)}が、著者自身も述べているように

- 被験者に課した試験問題の内容がAAの内容にふさわしくなかった
- AAの実行制御機能が貧弱であった

● 説明文を添付するなどの機能がなく、AA の説明力が不足であった
などの要因から実験の妥当性に疑問が残るものであった。したがって AA の教育的効果について判断するためには、調査の内容や環境を整備して再実験する必要があると思われる。まず当然のことながら、試験問題は AA 上に表現された知識を学習者がどの程度観察、獲得できるかを問うものに改める必要がある。また AA システムなどの対話型システムを援用してアルゴリズムを学習する場合、学習は教科書などの文書閲覧とシステムを用いた実験観察を繰り返しながら進行していくことが自然であろう。そういった環境において学習者は、教科書から体系化された理論を AA システムから直観的で経験的な知識を獲得することが期待される。したがって、この双方の表現媒体の間で内容や表現の一貫性を保持したり両者の対応関係を明示したりすることによって、学習をスムーズに進行させる工夫が求められる。これらのことは複数の媒体を別々に使用して学習することから生じる不利益を抑制することを狙いとす。

次に AA が持つ直観性と対話性の高さという特徴を活用して、より優れた学習環境を実現することについて考察する。多くの AA システムは、アルゴリズムにさまざまな入力を与えてその振る舞いの変化を確認できるシミュレータとして働く。この特徴は、学生実験などにおいてアルゴリズムの特徴を観察者に学習させる際に非常に有効であると思われる。また AA をその高い直観性を利用して拡張することはこれまでもよく行なわれてきている。その典型的な例が、複数の視覚的表現を比較参照することでより深い知識の理解を支援するというものである。この種の拡張法としては先述した MV や MA という 2 つの方法が従来から提案されている。MV は *Zeus* や *Tango* など代表的な AA システムの多くで実装されている機能で、ひとつのアルゴリズムを複数の観点から同時に観察できるようにし、そのさまざまな特性を立体的に理解することを支援するものである。ある観点から設計された視覚的表現が view であり、各 view には独立したウィンドウが割り当てられ

ることが多い。しかし学習教材としては、完全に独立した複数のウィンドウを適宜見比べるという方法は必ずしも適切ではない。例えば木の走査を AA にする際に、

W1. 訪問済みのノードをマークしていく

W2. スタックの内容を表示していく

という 2 種類の view を設計したとする。W1 には処理の概念的な側面が示され、W2 には実際の処理の特徴的な部分のひとつが示されることになる。これらの view を同時に観察したとしても、*Zeus*、*Tango* など従来のほとんどの MV では、W2 上でスタックに push/pop されたノードが W1 上のどの部分木の根なのかなど、2 つの view の相関関係を学習者が自ら判断しなければならない。複数の視覚的表現を組み合わせることによってより大きな効果が生じることは十分に期待できることから、組み合わせるための配慮を AA の設計に盛り込んだり、MV のために view を連繫させる機能をシステムに組み込んだりするなどして、view 間の相関関係を学習者に AA システムから提示する必要があると思われる。

一方 MA は複数の AA を同時に実行することにより、アルゴリズムを競争させたり動作の違いを観察したりするものである。この拡張は *Zeus* において実装されておりその有用性にも説得力はあるが、複数のアルゴリズムをどういった基準で同期させるかが非常に難しい。アルゴリズムの性能や部分問題への分割法、性能特性などアルゴリズムのどのような側面を比較するかによって AA の同期点も当然変化するからである。例えば選択ソートのアイデアを示すためのアニメーションでは、あえて最小 (大) 値を探索する過程は省略して値の移動と配列のシフトを中心に AA を設計することは自然である。しかし当然そのような AA をほかの整列アルゴリズムのアニメーションと比較してアルゴリズムの性能的な優劣を学習者に伝えることは非常に難しい。そういった目的のためには解析によって求められる性能に比例した長さには AA を作成することなどが必要となる。しかし複雑な処理を省略したり、逆に特徴的な部分を強調したりすることが AA 設計の大きな注力点のひとつ

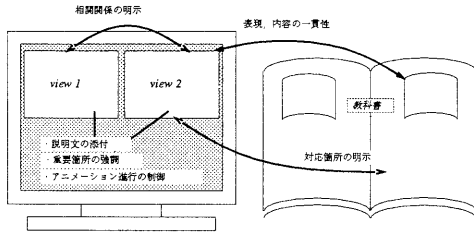


図2 AAを含んだ学習支援環境

Fig. 2 AA supported algorithm learning environment

つであるため、直観的な分かりやすさや自然な動きを保ったままアルゴリズムの性能に忠実な長さのAAを設計することは非常に困難である。また複数のAAを比較し易くするためには、アルゴリズムの動作の重要な相違点を強調することも求められよう。以上のようにMVに比べてMAは難しい問題が多く、本研究ではAAの直観性を活用するための方法としてMVを優先的に考慮することとした。

従来からAAシステムは対話性に優れていると主張されることが多かったが、実際にはほとんどのシステムにおいてアルゴリズムに与える初期入力をユーザが指定できたり、アニメーションの進行をVCRのように制御できたりするくらいの機能しか持たなかった。しかし学習を支援するためには、AAに表現されている知識を確実に学習者に伝達するために対話性をより向上させる必要がある。AAは直観性に優れる反面、

- AA上の視覚的オブジェクトやその振る舞いの意味
- AAが表現している知識の背景

などを厳密に観察者に伝達することは難しく、それらについて解説するための視覚的表現や文章、教科書での対応箇所などの情報を学習者の要求に合わせて提示する機能が必要であると思われる。またシステムから一方的に情報を提示するだけでなく、学習者からの問い合わせに対して情報を提供する機能も求められよう。これらのような対話性の向上は主に複数のviewを同時にディスプレイ上に表示することから生じる繁雑さを抑制する効果も期待される。図2に示した通り、本研究に

おけるAAの設計や拡張は、

- (1) 説明文の添付や重要な箇所の強調などの教育的配慮を行なうこと
- (2) 教科書との表現、内容の一貫性や対応箇所の明示などの統合的な環境を構築することなどによってAAの教材としての価値を高めることを目的としている。

2.2 AA設計の具体例

AAの設計についてより具体的に示すために本節では木の前順走査を例に述べる。教科書9)の該当ページ(pp.45)には、前順走査の規則、実装の方法、実装が正しいことの証明などについて記述されている。AAの設計にあたっては2つの指針を考慮するべきであると考えられる。まずひとつは教科書中の知識を直観的、動的に表現することである。ただしこの例題の場合、登場するおもなオブジェクトが木やスタックという抽象データ型であり、すでに視覚的表現法がある程度確立されている。そこで基本的なviewとして、抽象データ型のための2分木view(TV)とスタックview(SV)、および処理結果を提示するための訪問リストview(LV)の3種を用意することとする。もうひとつの指針は概念的な知識と実装上の知識を対応づけて学習者に提供することであり、この例題の場合こちらの指針の方がよほど重要である。例えばこのアルゴリズムにおいて、主題である木とそれを取り扱うための道具であるスタックという2種の抽象データ型は異なる抽象レベルに存在している。このため両者を単独に個別に表示しただけでは、それらの関係を学習者に伝えることは困難である。そこでプログラムコードの実行を追跡するためのプログラムコードview(CV)をviewに加える。以上に挙げた4種のviewの基本的な仕様を表1にまとめる。さらにこれらのviewを連繋して動作させることによって、view上に表現された知識の対応関係を提示することを考える。このアルゴリズムにおいては、push(), pop(), visit()という3つの操作を正しく適用していくことが本質である。そこで、あるノードがpush()される際にはTV上の対応するオブジェクトを複製してSV上のスタック上端に移動させ

表 1 木の走査のための view

Table 1 4 views for traversing tree algorithm

名 称	表現対象	表示物	動き
2分木 view (図 3-(左上))	処理の進行状況と規則性	初期入力の木	訪問済みノードを塗りつぶす
スタック view (図 3-(右上))	実装法の要所	バッファ用スタック	push/pop 操作時に更新
訪問リスト view (図 3-(下))	処理結果	訪問済みノードのリスト	新たにノードが訪問される度に更新
プログラムコード view	実装自身	プログラムコード	コードの実行中の部分を強調表示

るようなアニメーションを演出する。逆にノードが pop() される時には TV 上の対応オブジェクトをフラッシュし、連続して適用される visit() にタイミングを合わせて SV 上のスタック上端の要素を LV 上のリストの右端に移動させるアニメーションを作成する。これらはいずれも、view の組み合わせのための配慮に基づいた view の連携機能にあたる。また CV 上においては 3 種の操作が適用されるごとに、その引数や戻り値、簡単な説明文などを表示すればなお効果的であろう。

このようにあるアルゴリズムについてアニメーションを設計する際には、まずおもなオブジェクトと入出力、鍵になる操作群を抽出することから始めるべきである。次いで前 2 者を主題として各 view の外観を、後者から各 view の動きをそれぞれ定めることにより、良質な view を得ることが

できる。また先述したように、view の外観の決定にあたっては併用すべき教科書中の図などに同種のものがあれば、同様な視覚的表現を極力用いるべきである。個々の view について仕様が定められれば、次にそれらの連繫動作を設計する。その際には view 間のセマンティックギャップを埋めることを主眼として、view 上の視覚的オブジェクトや動きの対応関係を明示できるように設計する。そして view のさまざまな組み合わせを考慮しながら、適切な箇所を強調したり説明文を添付したりすることによってアニメーション全体の表現力を強化するのである。以上のように設計した AA を教科書と併用することにより、図 2 に示したような統合的なアルゴリズム学習環境を提供することが可能になり、その効果は従来の AA システムに比べ大きなものになると考えられる。

3. ま と め

本稿では、従来開発されてきた多くの AA システムが必ずしも実用的とはいえないことを述べ、より应用到即した形態で AA 技術を評価すべきであることを主張した。このときもっとも重要な問題は、AA は直観的な表現ではあるが知識を厳密に表現して観察者に伝達することにはあまり向いていないことである。したがって解説文を添付したり教科書を参照したりすることが必要となり、AA の内容も独自のものにすることを避けて教科書の内容や表現と統一性を保つように作成することが求められる。一方 MV のように AA の表現力を強化することも重要であるが、MV はかえっ

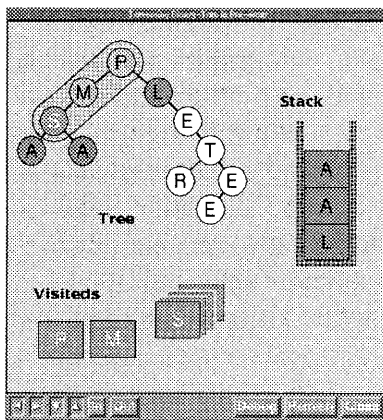


図 3 各 view の画面

Fig. 3 cooperative multiple views

て混乱を招く危険性も大きい。したがって各 view の一部の内容や動きについて予想される組み合わせに沿ったいくつかのパターンを用意し、観察者が指定する view の組み合わせに対して適切な内容に自動的に切り替わるような機構がシステムに必要となろう。また AA は対話型シミュレータでもあるので、

- view を組み合わせる
- 初期入力を変える
- 情報をシステムに問い合わせる

などの能動的な学習により効果を発揮すると思われる。今後はこれらの考察に基づいた AA システムを実装し、実際に評価実験を行なって AA の可能性について研究を進めていきたい。

参 考 文 献

- 1) Badre, A., Beranek, M., Morris, J.M. and Stasko, J. T.: Assessing Program Visualization Systems as Instructional Aids, Technical Report GIT-GVU-91-23, Graphics, Visualization, and Usability Center, College of Computing, Georgia Institute of Technology, Atlanta, GA 30332-0280 (1991).
- 2) Baecker, R. M. and Sherman, D.: Sorting Out Sorting, Shown at SIGGRAPH '81, Dallas, TX, 16mm color sound film (1981).
- 3) Brown, M. H.: Zeus: A System for Algorithm Animation and Multi-View Editing, in *Proceedings of the 1991 IEEE Workshop on Visual Languages*, pp. 4-9 (1991).
- 4) Ford, L.: Goofy Animation System, "Research Report" 266, Department of Computer Science, University of Exeter, Exeter, U.K. (1993).
- 5) Gloor, P. A.: AACE - Algorithm Animation for Computer Science Education, in *Proceedings of the 1992 IEEE Workshop on Visual Languages*, pp. 25-31 (1992).
- 6) Lawrence, A.W., Badre, A.M. and Stasko, J. T.: Empirically Evaluating the Use of Animations to Teach Algorithms, in *Proceedings of the 1994 IEEE Symposium on Visual Languages*, pp. 48-54 (1994).
- 7) Naps, T. L. and Swander, B.: An Object-Oriented Approach to Algorithm Visualization Easy, Extensible, and Dynamic, *Papers of 25th SIGCSE Technical Symposium on Computer Science Education*, pp. 46-50 (1994).
- 8) Roman, G.-C.: Pavane: A System for Declarative Visualization of Concurrent Computations, *Journal of Visual Languages and Computing*, Vol. 3, No. 2, pp. 161-193 (1992).
- 9) Sedgewick, R.: *Algorithms*, Addison-Wesley, 2nd edition (1988).
- 10) Stasko, J. T.: Tango: A Framework and System for Algorithm Animation, *IEEE Computer*, Vol. 23, No. 9, pp. 27-39 (1990).
- 11) Stasko, J. T., Badre, A. and Lewis, C.: Do Algorithm Animations Assist Learning? An Empirical Study and Analysis, in *Proceedings of the ACM INTERCHI '93 Conference on Human Factors in Computing Systems*, pp. 61-66 (1993).
- 12) Takahashi, S., Miyashita, K., Matsuoka, S. and Yonezawa, A.: A Framework for Constructing Animations via Declarative Mapping Rules, in *Proceedings of the 1994 IEEE Symposium on Visual Languages*, pp. 314-322 (1994).
- 13) 高橋, 松岡, 宮下, 米澤: アルゴリズムアニメーション作成システムにおける宣言的記述方法について, in *Proceedings of インタラクティブシステムとソフトウェア I*, 日本ソフトウェア科学会 WISS '93, pp. 161-168 (1993).