LK-019

グラフ上の動画作成を可能にする動画記述言語と動画生成エンジンの設計および実現

A design and realization of a discriptive language and generating engine for animations on graphs

佐々木 敏晃[†] 片山 喜章[‡] 高橋 直久[§]
Toshiaki Sasaki Yoshiaki Katayama Naohisa Takahashi

1. はじめに

本稿ではグラフの上で物が動いたり変形したりする動画を作成することが可能な動画記述言語と動画生成エンジンを提案する．

事象の視覚的な手助けとして，事象の連続的な変化の様子を表現することが可能な動画 (アニメーション) は有効な手段である [1]．例えば，地図を用いたナビゲーションシステムやグラフに関するアルゴリズムなどはユーザに対して道筋や動作を伝えるのに非常に有効である．ナビゲーションシステムでは移動経路情報がグラフに当たり，ナビゲート対象者がグラフ上を移動する．またグラフ上でのアルゴリズムの１つである分散アルゴリズムでは分散ネットワークがグラフに当たり，メッセージがグラフ上を移動する [2]．このように，単純な図形のみからなるグラフ上の動画はさまざまな適用範囲が考えられ有効である．

一般的な動画を作成する方法としてセル画方式とオブジェクト方式の２種類ある．方式１のセル画方式では動画を構成しているセル画 (フレーム) を一枚一枚描画することで動画の作成を行なっている．一方，方式２のオブジェクト方式では動画に登場するオブジェクトに関してそれぞれ動作を記述することで動画の作成を行なう．両方式の利点と欠点を以下に示す．

方式 1．セル画方式

利点 特殊な再生 (早送り，スロー，逆再生等) が容易．任意の図形のアニメーションが可能．

欠点 作成が大変．全てのフレームを保持するため記憶容量が多くなる．動作内容の変更が困難．

方式 2．オブジェクト方式

利点 作成が簡単．オブジェクトとそれらに対する動画命令のみを保持するため，記憶容量が少なくてすむ．動作内容の変更が容易．

欠点 特殊な再生が困難．単純な図形のみしか扱えない．

グラフを利用した動画は単純な図形のみから構成されることが多い．そのため，グラフを利用した動画の作成にオブジェクト方式が良く使われる．オブジェクト方式を用いた動画を作成するツールとして PowerPoint がある [3]．PowerPoint ではオブジェクトを配置し，それぞれのオブジェクトの動作内容 (出現，消滅，移動等) を設定することで動画を生成している．この方式ではオブジェクトで表現できる動画を直感的かつ簡単に記述することが可能であるが，次の欠点をもつ．作成したアニメーションは特定のアプリケーションでしか利用できない．また，セル画方式に比べ，オブジェクト方式は特殊な再生や再生品質の変更を簡単な方法で実行することが難しい．

そこで本稿では記述が容易なオブジェクト方式と特殊な再生が容易に行なえるセル画方式の両方の利点を活かした，グラフ上の動作を対象とする動画記述言語と動画生成エンジンを提案する．提案システムの主な機能を次に示す．

機能 1 システム内で実時間とは独立に管理される時間 (仮想時間) を導入し，仮想時間と実時間を任意に対応付ける

機能 2 任意の仮想時刻のオブジェクトの属性 (表示属性，色，移動速度等) を求める

機能 3 入力命令 (順再生用命令) から逆再生用命令を生成する

機能 4 任意の仮想時刻のオブジェクトの属性 (仮想フレーム) からビットマップ画像を生成する

本システムでは，オブジェクト方式で動画命令を記述し，再生時には動画生成エンジンによってセル画方式で動画を生成する．また，本システムにおけるオブジェクト方式での動画とは，オブジェクトに対してそれらの持つ属性の値 (表示位置，色，形，速度など) を変化させることによって動きを表現する方式と考える．

一般にユーザが動画を作成する際には，オブジェクトの動きの指定時には実時間を意識せず，動きの順序，つまり各動きの間隔による相対時間で指定すると考えられる．そこで，本システムでの動画記述命令はユーザがオブジェクトの動きを時間順に書き下していく形式を採用する．具体的には，ユーザがオブジェクトの動きを指定する (属性値を変化させる) 際に，そのタイミングを相対的な時刻で指定する．この時間を「仮想時間」と呼ぶ．仮想時間の導入により，ユーザは実時間を意識することなく，頭の中での動画イメージを直観的に動画命令として書き下すことが可能になる (図 1)．文献 [4] ではユーザーの経験時間とメディアの時間の概念を導入することで，時間的な連続性を持つデータの概観と詳細の表現と，その表現との時間的なインタラクションが可能であることが述べられている．本稿の手法は，文献 [4] と同様な時間的操作をグラフ上の動画に施す機能を実現するための基礎となる．さらにユーザが記述した動画命令は，動画生成エンジン内で仮想時間の経過にしたがって解釈される．つまり，仮想時間の経過とともに各オブジェクトの属性値が連続的に変化していく．このとき，任意の仮想時刻での各オブジェクトおよびそれらの属性の集合を「仮想フレーム」と呼ぶ．仮想フレームは最終的には動画

[†]名古屋工業大学大学院 情報工学専攻
[‡]名古屋工業大学大学院工学研究科 おもひ領域
[§]名古屋工業大学大学院工学研究科 ながれ領域

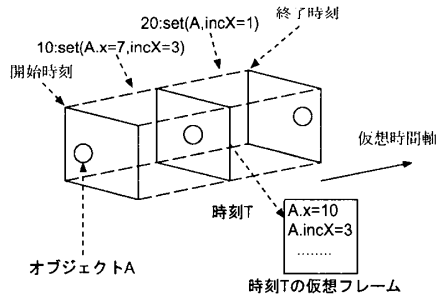

図 1: 仮想時間と仮想フレームの概念図

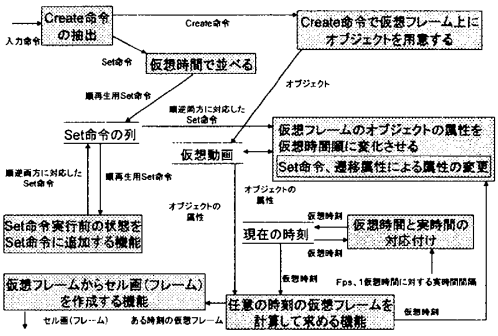

図 2: 提案システムの全体構成

生成エンジン内でビットマップ画像に変換され，動画を構成するフレーム (セル画) としてユーザに提示される．

以下に，本システムの特徴について述べる．

特徴 1 任意の速度で再生が可能 (機能 1 より)
機能 1 により，(ユーザが指定する) 動画生成エンジン内の仮想時間と実時間が対応づけられ，実時間上での動画再生が可能になる．この際，対応づけの割合を自由に設定できるため，動画再生の速度を任意に設定可能になる．

特徴 2 ほぼ一定のメモリ量で単位時間に表示するフレーム数 (fps) を自由に設定可能 (機能 1,2 より)
機能 2 によって，任意の仮想時間間隔で仮想フレームが取り出せる．つまり，実時間と仮想時間の対応づけを変化させることなく仮想フレームを取り出す仮想時間間隔を変化させることで，単位実時間あたりに生成 (表示) するフレーム数を自由に設定可能になる．この際，フレームを前もって生成する必要がないことから，再生に必要な記憶領域は fps の影響を受けない．

特徴 3 少ないメモリ量で逆再生可能 (機能 3) 一般に逆再生は，あらかじめ動画全体を通してフレームを用意しておき，それらを逆順に表示させることで実現する．それに対し本システムでは，ユーザからの動画命令 (順再生用) を読み込む際に同時に逆再生用動画命令を生成し埋め込む．したがって，逆再生も順再生と全く同様に扱うことが可能になる．

特徴 4 アプリケーションによらない再生が可能 (機能 4)
機能 4 により，本システムからの出力 (動画) における各フレームは，一般的なビットマップ画像である．これらを MNG, アニメーション GIF にすることで，それらが表示可能なアプリケーションであれば，本システムで生成された動画が再生可能になる．

本稿の以降の構成は次の通りである．2 章で提案システムの設計および実現，続く 3 章で提案システムの機能および特徴の実証実験について述べ，最後の 4 章でまとめを行ない今後の課題を明らかにする．

2. 提案システムの設計および実現

本提案システムではユーザーから入力として，動画内容の示す命令記述と動画操作を受けつけ，それに対して動画内容を示す連続したビットマップ画像を出力する．提案システムの全体構成に付いて図 2 に示す．

2.1 動画記述言語の設計

提案システムにおけるグラフ上での動画を記述するための動画記述言語について述べる．

提案する記述言語で利用する命令は，表 1 のとおりである．このうち，create 命令でオブジェクトを用意し，set 命令を用いてオブジェクトの属性を設定することで動きを指定する．また System オブジェクトの属性を設定することで，再生に関する各種パラメータを指定する．また，命令によって扱えるオブジェクトとその属性は表 2 と表 3 のとおりである．

ユーザーは，これらの命令の列 (動画命令列) を記述する際に，set 命令の前に，命令間の相対的な時間，つまり命令を実行する仮想的な時刻を指定する為の「仮想時刻」を指定する (例 100:set(A,x=10))．set 命令によって指定される属性は，基本的には指定された仮想時刻に瞬時に設定され，それ意外の時間では変更しない．しかし，属性の中に用意された「遷移属性」に関しては，これによって指定された属性が仮想時間の経過とともに連続的に変化する．たとえば，属性 incX の値を 2 と設定した場合，単位仮想時間ごとに属性 x の値が 2 ずつ増加する．この遷移属性の指定によって，オブジェクトの連続的な動作が指定可能になる．

2.2 提案システムの概要

提案システムが入力を受けてから動画を出力までの処理について，その概要について述べる．ユーザが記述した動画命令列 (入力動画命令列) と再生に関するパラメータが提案システムに入力されると，動画命令列および再生パラメータにしたがって動画 (ビットマップ画像の系列) が出力される．この際，提案システム内では，以下の処理が行われる．

処理 1-1 入力動画命令列から create 命令と set 命令を分離し，すべての create 命令を命令列の先頭部分に集める．

処理 1-2 分類した set 命令を命令の仮想時刻順に並び換える．

処理 1-3 分類した create 命令を解釈することで，仮想時刻 0 におけるオブジェクトとその属性の集合 (初期仮想フレーム) を作成する．

処理 1-4 初期仮想フレームと仮想時刻順に並び変えた set 命令を使用して，set 命令に新に逆再生用の項目を加える．

表 1: 提供する動画記述言語の命令セット

命令名	引数
create	オブジェクトの種類，識別子
set	識別子，属性の設定記述 (属性名=設定値)

表 2: システムで提供するオブジェクトの種類

種類	説明
Plane(平面)	平面を描画する属性をもつオブジェクト
Line(直線)	直線を描画する属性をもつオブジェクト
String(文字列)	文字列を描画する属性をもつオブジェクト
System(システム)	再生状態に関する属性をもつオブジェクト

以上の処理 1-1〜1-4 により，ユーザから入力されれた入力動画命令列を提案システムで再生可能 (動画出力可能) な動画命令列に変換する．変換された動画命令列と再生パラメータ，さらに処理 1-3 で得られた初期仮想フレームを用いて以下の処理を行うことで，動画 (ビットマップ画像 (フレーム) の系列) を出力する．

処理 2-1 再生パラメータから得られたフレーム生成間隔をから，現仮想時刻の次にフレームを生成すべき仮想時刻を求める．

処理 2-2 求た仮想時刻における仮想フレームを生成する．

処理 2-3 得られた仮想フレームの情報から，各オブジェクトをビットマップとして描画したフレームを生成する．

以上の処理 2-1〜2-3 を繰り返すことにより，動画を出力する．

本稿のスペースの都合より図 2 中の「仮想時間と実時間の対応」,「任意仮想時刻における仮想フレームの生成」および「逆再生用命令の付加」についてのみ説明する．

2.3 仮想時間と実時間の対応

提案システムでは，ユーザが指定しシステム内で扱う仮想的時間と，実時間との対応を自由に変更することで再生速度の任意の変更が可能となっている．

仮想時間と実時間の対応のために，以下の二つの入力をとる．

入力 1-1 単位時間当たりに表示するフレーム数 (fps)

入力 1-2 単位仮想時間に対する実時間間隔

入力 1-1，1-2 より，仮想フレームを生成すべき仮想時間間隔を求めることが可能となり，したがって再生速度と動画再生品質を自由に設定できる．

2.4 任意仮想時刻における仮想フレームの生成

仮想時間軸に沿って連続的に変化するオブジェクトの属性値を，任意の仮想時刻を指定することでその時点での属性値の集合，すなわち仮想フレームを生成する機能について述べる．本機能は，以下の二つの入力をとる．

入力 2-1 処理 1-1〜1-4 によって得られた動画命令列

入力 2-2 ある仮想時刻とその時刻での仮想フレーム

入力 2-3 仮想フレームを生成したい任意の仮想時刻

提案システム内では，仮想時間軸に沿って，ユーザが指定した仮想時刻に各 set 命令が実行されている．各オブジェクトは set 命令によってその属性値を変化させる．また，遷移属性を指定されているオブジェクトは，連続的に属性値が変化している．このような状況で，指定された仮想時刻における仮想フレームを求めるために，以下の処理を行う．

表 3: 提案システムで提供するオブジェクトの属性

属性	説明
x，y，width，height	外見の大きさと配置場所
thickness，blnEgde	外見の外枠の太さと表示の有無
drawColor，fillColor	外見と外枠の色
shape	外見の種類
string，size	文字列の内容とフォントの大きさ
blnVisible	オブジェクトの表示の有無
incX，incY，incWidth，incHeight	外見の大きさと配置場所を変更する
blnChange	incX 等による変更の有無
zorder	オブジェクトが重なった場合の表示順番
inverval，fps	表示間隔と表示枚数

処理 3-1 入力 2-2 で与えられた時刻を始点時刻とし，始点時刻と入力 2-3 で与えられた時刻を変更先時刻とするとき，始点時刻と変更先時刻から進行方向 (順方向，逆方向) を求める．

処理 3-2-1 入力 2-1 で与えられた動画命令列において，進行方向上で始点時刻からもっとも近い set 命令を取り出す．該当する set 命令がない場合，あるいはその set 命令が変更先時刻を越えている場合は，処理 3-3 を行う．

処理 3-2-2 遷移属性が有効な場合 (始点時刻と処理 3-2-1 で取り出した set 命令の仮想時刻の間以前に遷移属性が指定されている場合) は，遷移属性を用いて当該属性の値 をその間に相当する分だけ変化させる．

処理 3-2-3 処理 3-2-1 で取り出した set 命令を実行し，属性値を変化させる．同時にその時刻を始点時刻とし，処理 3-2-1 へ戻る．

処理 3-3 遷移属性が有効な場合，始点時刻から変更先時刻の間の分，遷移属性に従って属性値を変化させる．

上記処理を行うことで，任意の変更先時刻時点における仮想フレームを生成することができる．

2.5 逆再生用命令の付加

提案システムでは，逆再生の際にもあらかじめフレームを用意することなく，動画命令列から必要時に仮想フレームを取り出し，フレームを生成する．

本機能は，以下の二つの入力をとる．

入力 3-1 処理 1-1〜1-4 によって得られた動画命令列

入力 3-2 初期仮想フレーム

入力 3-1,3-2 より，逆再生に必要な属性値を求め各 set 命令に付加することにより，逆再生が可能な動画命令列を生成する．具体的には，set 命令の解釈時に指定どおりに属性値を変化させる以外に，以下の処理を追加する．

処理 4-1 set 命令から，命令によって変更する属性名と設定値を取り出す．

処理 4-2 求めた属性名に対し，変化させる前の属性値を仮想フレームから求め，逆再生用項目として当該 set 命令に追加する．

処理 4-3 求めた属性名に対する属性値を，設定値に変化させる．

以上の処理について，例をあげて説明する (図 3)．以下，手順を示す．

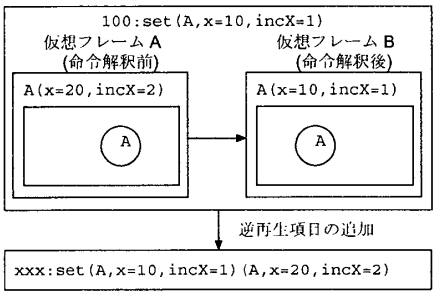

図 3: 逆再生項目の追加例

表 4: 再生品質によるメモリ使用量

フレーム間隔 (ms)	10	20	30	40
逆再生無し：メモリ使用量 (Mbyte)	13	13	13	13
逆再生有り：メモリ使用量 (Mbyte)	13	13	13	13

手順 1 変化前の仮想フレーム A から x=20，incX=2 を取り出す．

手順 2 set 命令の逆再生用の項目として取り出した値を付加する．

手順 3 set 命令を解釈してオブジェクトを変化させ，仮想フレーム B の状態にする．

上記の手順で逆再生用の項目を設定し，逆再生時に順方向と同様の処理で仮想フレームを生成する．

3. 実験

提案システムのプロトタイプを作成し，提案システムの特徴の一つである，自由な動画品質の設定について，機能確認とその結果に対する考察を行った．実験に使用した環境は，以下のとおりである．

- OS：WindowsXP Professional
- CPU：Celeron 1．1GHz
- 記憶容量：256Mbyte
- 開発言語：JAVA(JSE1．4．1_02)
- 測定ツール：WindowsXP 付属タスクマネージャ

本実験では，提案システムに与えるフレーム表示間隔をさまざまに変更することで，実際に動画の再生品質が制御可能かどうかを目視により確認する．同時に，再生時のメモリ使用量を観測することで，再生品質と必要メモリ量の関係を明らかにする．

測定に使用した動画は，ライン上に配置したノード上をメッセージが移動するものである (図 4)．この動画命令列に対して、逆再生項目を追加したものと順再生のみのもの用意し、再生パラメータとして，1 仮想時間当たりの実時間間隔を 1ms，またフレーム表示実時間間隔を 10ms から 10ms 単位で 40ms まで変化させた．

実験の結果，フレーム表示間隔を変化させることで，実際に表示される動画の品質が荒くなったり滑らかになることと、逆方向の再生を確認した．また，使用メモリと表示間隔の関係は，表 4 のようになった．以上より，本実験において仮想フレームの導入による効果として，動画の品質に対し一定のメモリ量で変更が可能であることと，順再生とほとんど変わらないメモリ量で逆再生可能であることがわかった．

```
create(1,Plane)
create(2,Plane)
create(3,Plane)
create(4,Line)
create(5,Line)
create(6,Plane)
0:set(1,x=10.0,y=285.0,width=50.0,
      height=30.0,blnVisible=true)
0:set(2,x=275.0,y=285.0,width=50.0,
      height=30.0,blnVisible=true)
0:set(3,x=540.0,y=285.0,width=50.0,
      height=30.0,blnVisible=true)
0:set(4,x=35.0,y=300.0,width=265.0,
      height=0.0,blnVisible=true)
0:set(5,x=300.0,y=300.0,width=265.0,
      height=0.0,blnVisible=true)
0:set(6,zorder=23,x=25.0,y=290.0,
      width=20.0,height=20.0,blnVisible=true)
0:set(6,incX=2.65,incY=0.0,blnChange=true)
100:set(6,incX=2.65,incY=0.0,blnChange=true)
200:set(blnChange=false)
```

図 4: 実験で用いた動画記述命令

4. まとめと今後の課題

本稿では，グラフ上で単純な図形が移動・変形するような動画を作成可能な動画記述言語と動画生成エンジンを提案し，実装を行なった．その結果，提案システムはオブジェクト方式による直感的かつ容易な動画記述と，セル画方式による自由度の高い再生が可能であることを確認した．また，本稿では述べなかった，提案システムを分散アルゴリズムアニメーション記述システムとして利用するためのトランスレータを作成 [5] し，同等の動画を記述する際に PowerPoint に比べて動画作成時の手間の少ないことを確認した．また，トランスレータの作成も，若干の手間はかかるものの作業自体は非常に容易であることも確認している．このシステムは，出力先として PowerPoint のアニメーション形式が指定できる．このことより，出力先を選ばないという特徴も明らかになったと考える．

今後の課題としては，残りの特徴を定性的あるいは定量的に確認するための実験が必要と考える．さらに，本稿では詳しく述べなかった動画としての表現力の向上も課題のひとつである．

謝辞　本稿に対し大変有益な御意見をいただいた査読者の方々に感謝の意を表す．

参考文献

[1] 小池 英樹：bit 別冊 ビジュアルインタフェース -ポスト GUI を目指して-，平川，安村偏，第 2．1 章，pp．24-44，共立出版，1996．

[2] 亀田恒彦，山下雅史：分散アルゴリズム，近代科学社，1994．

[3] Microsoft Office PowerPoint2003 http://www.microsoft.com/japan/office/powerpoint/prodinfo/default.mspx

[4] 高嶋章雄，山本恭裕，中小路久美代：検索的データ分析のための時間的な概観と詳細の表現およびインタラクションに関する研究，情報処理学会論文誌，Vol. 44，No．11，pp．2767-2777，November，2003

[5] 佐々木 敏晃：汎用的動画生成エンジンの実現と分散アルゴリズムアニメーション作成ツールへの適用，名古屋工業大学 電気情報工学科卒業論文 (2004-3)