

物理的特徴に基づく擬音語可視化手法の検討

福里 司^{†1} 森島繁生^{†2}

本稿では、物理パラメータを基に擬音語を推定する手法を検討する。擬音語とは、アニメや漫画作品に登場するキャラクターの質感や動きを言語化したものであり、受け手に内容や状況を直感的に認識させる表現技法である。しかしアニメーターが、状況や設定資料を基に感覚的に決定しているのが現状である。そこで、CGアニメーションの制作において必要となる複数のパラメータを基に、擬音語を自動推定及び付与する手法を提案する。この技術により、映像のみでは把握できない環境情報(素材感や速度感など)を可視化することが可能となる。

Estimation of Onomatopoeia Considering Physical Phenomena

TSUKASA FUKUSATO^{†1} SHIGEO MORISHIMA^{†2}

This paper presents a technology that enables to estimate onomatopoeias based on physical parameters. Onomatopoeias often are used to emphasize material and character's motion in Non-photorealistic Animation (NPR), and enable users to understand anime contents intuitively. However, animators empirically select onomatopoeia according to the situations and character references. Therefore, we propose a method to quantify and recommend onomatopoeia using physical parameters which is computed within the animation process. This makes it possible to visualize the information (material and speed etc.).

1. はじめに

アニメや漫画には、キャラクターの質感、動きを強調するために、擬音語が頻繁に用いられている。擬音語とは音声や動きを言語表現化したものであり、受け手に内容や状態を直接的に訴えかけることができるため、ダイナミックさを演出する際や、内容を把握させるための補助ツールとして用いられる。近年ではアニメコンテンツだけでなく、新聞や雑誌などの活字媒体や宣伝広告、俳句や絵本などといった文芸作品など幅広い分野で頻繁に利用され、世界的にも注目されている。しかし最適な擬音語を選択し、アニメーションを制作するには、2つの課題がある。1つ目として、アニメ作品には現実と大きく異なるシーンが多数存在し、実際の音では表現することができないこと、2つ目として、アニメの制作現場では、完成した映像と別途で音声を録音する After Recording と呼ばれる手法が一般的に用いられ、映像中の音声を基に擬音語を決定するような状況が極めて少ないことが挙げられる。そのような背景から、アニメーターや漫画家にはキャラクターの動作や設定資料に記載される情報のみを基に、経験的に決定する技術が要求されてしまい、アマチュアのアニメーターが任意の場面に応じ、最適な擬音語を付与したアニメーションを制作することは困難と言える。

そこで本稿では、CGアニメーションの制作において必要となる複数の物理パラメータ（動的パラメータ及び静的

パラメータ）を基に擬音語を自動で推定し、映像中に付与する手法を提案する。提案手法の着眼点は、漫画家やアニメーターが擬音語アニメーションを制作する際に利用する「擬音語から感じ取れる速度感や粘性感などの印象は、擬音語を構成する文字(アルファベット)によって決定される。」という経験的な法則を定式化することにある。本稿の研究プロセスは以下に示す。

1. 事前に簡易的なアニメーションを鑑賞し、その状況にあった擬音語を主観的に選択する。
2. 簡易的なアニメーション上で算出される物理的パラメータを、アルファベット1文字ごとに分離し、回帰モデルを生成する。
3. 任意のCGアニメーション中で算出された物理パラメータを基に、最適な擬音語を推定する。
4. 擬音語を3D空間上にマッピングする。ただし、推定された擬音語同士が重なって描画されないように制約を設ける。

それにより、CGアニメーションを制作する工程で、設定資料に記載されている物理的な特徴(物理パラメータ)を入力することで、状況に合った最適な擬音語と、擬音語を描画するための最適な位置を、高速に提示することができる。また、擬音語の推定結果の上位を提示し、編集及び再学習機能を付加することで、各自の好みに応じた擬音語付きアニメーションを簡易的に生成することができる。本技術によって、アニメ作品の制作において、最も人手を要する要素の一つを効率化し、アマチュアのアニメ作品の制作支援につながると考えられる。また、擬音語を付与することで映像中の情報を補い、一目で内容を把握するための映像要約技術としても応用が期待される。

^{†1} 早稲田大学/JST
Waseda University/JST
Email: tsukasa@moegi.waseda.jp

^{†2} 早稲田大学理工学術院/JST
Waseda University/JST
Email: Shigeo@waseda.jp

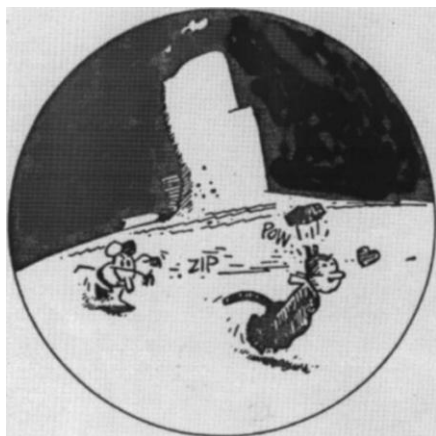


図 1 衝突現象における擬音語アニメーションの一例
from George Herriman's "Krazy Kat", 1913.

2. 関連研究

アニメや漫画特有の技法に着目した手法は多数提案されている。1 つ目として、キーフレームアニメーションを入力とし、効果線やモーションブラー^{[1][2]}、Dynamic Glyph^[3]を生成する手法が挙げられる。しかし、これらは時系列的な情報を基に、3D モデルを再生成する必要があることや、フレーム間の解析に処理時間を要する問題があった。そこで Umeda らは、Kinect で取得した関節情報を基にリアルタイムにエフェクトを付与する技術を提案した^[4]。しかし、この手法は人間の既定の動作に対し、一律の効果線や文字を描画するものであり、アニメや漫画のように骨格推定できないようなキャラクタに対応することは困難である。

一方、物理シミュレーションの過程で計算されるパラメータを基にエフェクトを生成するアプローチとして、土橋らはサウンドテキストチャを用いたリアルタイムな風切音を実現した^[5]。しかし、この手法は物体の形状と速度のみで決定されるものであり、素材感の反映ができない。よってアニメ作品に登場するような素材感を反映した擬音語を決定することは困難である。さらに近年、音声空力場の計算によって生成する手法^[6]も提案されているが、音声と擬音語は必ずしも対応づけることができないため、本目的を達成することは困難である。

擬音語に関する研究として、音響特徴量を基に擬音語を判定^{[7][8]}、および画像特徴量を基に可視化する手法があげられる^[9]。この手法によって、動画に自然かつダイナミックな演出の可視化を可能とした。しかし擬音語の判定部は、音響情報のクラスタリング(タグの付与)であり、アニメーターが考える大量の擬音語の数に、音声分類をすることは現実的ではない。

さらに擬音語の定量化に関する研究として、大量の擬音語を主観評価実験によって、素材感や印象を定量化する手法が提案されている^{[10]~[13]}。しかし、この手法には定量化

が可能な擬音語に制限があること(例: もこもこ(XYXY 型)), 擬音の印象に左右するアルファベットが経験的に決定していること、解析処理に非常に人的及び時間的な労力がかかることが問題として挙げられる。

アニメ作品には、素材感や速度感などが存在していない場合が多く、その情報を補うために擬音語というツールが多用されている。しかし効果線の定量化^[14]と異なり、擬音語に着目した研究は、素材感などアニメーターが擬音語を決定するために用いる情報を反映できず、上記の手法を組み合わせても、擬音語を推定することは困難である。また作品ごとの擬音語の分散が大きく、映像内容ごとにモデル化が必要となることや、処理時間も膨大となり、現実的ではない。

そこで、我々は CG アニメーションの制作において必要となる物理的なパラメータを基に、擬音語を自動推定及び付与する手法を提案する。手順として、ユーザが簡易的なアニメーションを鑑賞し、映像中に付与したい擬音語を選択し、アニメーションを生成するために算出される物理パラメータの対応を学習する第 1 ステップと、任意の物理シミュレーションやキーフレームアニメーションの生成過程で算出される物理パラメータを基に、擬音語データベースから検索する第 2 ステップの 2 段階構成となっている。この技術により映像コンテンツ内の素材感や速度感などを強調し、より見ごたえのある映像を生成することが可能となった。また本手法は、高速に擬音語の提示及び描画を行うことができるため、よりユーザの好みに応じた編集も可能となった。本アルゴリズムは、擬音語と物理パラメータの対応関係を学習し、その結果を基に自動推定する手法であるため、アニメーションで用いた物理パラメータの種類に制限はない。つまり擬音語を画像特徴量や音響情報を基に定量化することも可能であると考えられる。

3. 擬音語の学習・推定

アニメや漫画作品中で擬音語が出現するシーンは様々あるが、本研究の目的は、アニメーターが使用する擬音語を定量的に解析し、擬音語の経験的な法則を定式化することにある。そこで本稿では、擬音語のバリエーションが最も豊富であり、擬音語が付与されることが最も多い「キャラクタと物体の衝突アニメーション(図 1)」を対象とし、衝突に応じた物体の挙動とキャラクタの動きを表現する弾性体シミュレーション^{[15][16]}とスキニングを基にしたキーフレームアニメーションを制作した。

3.1 章で主観的に決定した擬音語と物理パラメータの対応関係の学習、3.2 章では学習データを基に擬音語を最適化する推定、3.3 章は 3 次元空間上に擬音語をマッピングする位置推定の手順について述べる。図 2 に本章で述べる処理の概要図を示す。

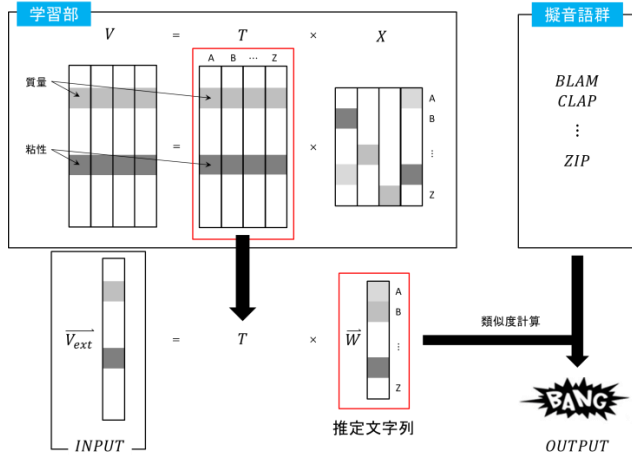


図2 システム概要

3.1 擬音語の回帰モデル生成

本章ではアニメーション上で必要となる物理パラメータと最適な擬音語を決定する手法を説明する。擬音語の印象を定量化するために、アニメーターや漫画家を用いる「擬音語から読み取れる粘性や質量、加速度、体積の印象は、構成する1文字（アルファベット）に依存する」という法則に着目する。Komatsu らはこの関係に着目し大量の擬音語と印象の対応関係を基に、2 モーラの擬音語の印象を定量化したが、対応付けや学習に非常に手間が掛かる上、印象に依存する文字(アルファベット)を経験的に決定しているため、選出しなかった文字にパラメータを付与することができない。

そこで本稿では、簡易的なアニメーションをユーザに提示し、その状況にふさわしい擬音語をデータベースから選択させる実験を行う。簡易的なアニメーションで用いられた物理パラメータを基に、擬音語と物理パラメータの対応付けを学習する。この作業を数回行い、各ユーザが考えるアルファベット1文字あたりに対応づいた物理パラメータを決定する。擬音語のデータベースとして、コミック販売部数及び、DVD 年間売上ランキングで上位の漫画・アニメ作品10作品を基に、衝突や衝撃に対する100種類の擬音語を用意した。擬音語の文字列(アルファベットの文字構成)はASCIIコード(D個)の成分を持つ列ベクトル $\mathbf{X}$ (図3、式(1))を用いる。

$$\bar{\mathbf{X}}_i = (x_0, x_1, \dots, x_D)^T \quad (1)$$

擬音語と物理パラメータの対応付けを k 回を行う。以上の手順により分類された k 種類の擬音語の文字列を以下の1つの行列で表すことができる。(本稿では実験的に $k=10$ or 20 とする。)

$$\mathbf{X} = (\bar{\mathbf{X}}_0, \bar{\mathbf{X}}_1, \dots, \bar{\mathbf{X}}_k) \quad (2)$$

第2ステップとして、任意の擬音語に対応付けられた物理

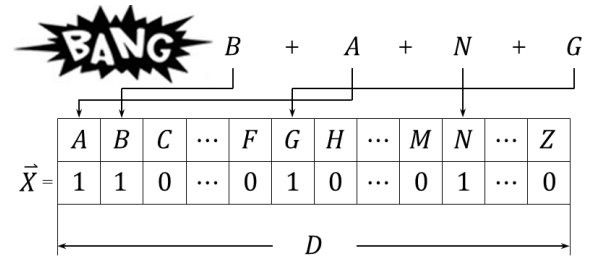


図3 文字列ベクトル $\mathbf{X}$ の表記方法

図4 印象パラメータ $\mathbf{V}$ の表記方法

パラメータをベクトルに適合することによって、異なる擬音語どうしの解析が容易に行える。本稿で述べる物理パラメータとは、アニメーターが擬音語を決定する際に参考にすると思われる素材感（質量、体積、粘性）や挙動（速度）を指す。(今回はパラメータ数を $n=5$ とする。) 選択した物理パラメータは擬音語を定量化する先行研究(Komatsu らの手法)と同様である。ただし本稿では、物体同士の衝突アニメーションを対する擬音語を対象としているため、衝突する2つの物体のパラメータを入力パラメータとして設定する。本稿で用いたパラメータを表1、物理パラメータをベクトル $\mathbf{V}_i$ に適合させる。簡易的なアニメーションに対し、適切な擬音語 i を選択する際、複数のアニメーターの主観評価実験によって、共通の法則(アルファベット1文字あたりのパラメータ)を定式化するのが望ましいが、今回は筆者1名による実験によって決定した。各理パラメータを列ベクトルの各成分に格納することで印象ベクトル $\mathbf{V}_i$ を式(3)のように表すことができる(図4)。

$$\bar{\mathbf{V}}_i = (v_0, v_1, \dots, v_n)^T \quad (3)$$

複数の擬音語の印象ベクトルを平均と分散を基に正規化し、ベクトル $\mathbf{V}_i$ を再度定義する。これにより k 種類の擬音語の物理パラメータを行列で以下の一つの行列で表すことができる。

$$\mathbf{V} = (\bar{\mathbf{V}}_0, \bar{\mathbf{V}}_1, \dots, \bar{\mathbf{V}}_k) \quad (4)$$

第3ステップとして、式(2)、(4)を基に擬音語に付与された物理パラメータを1文字ごとに分解し、回帰モデル $\mathbf{T}$ を生成する。分解結果は主観評価に依存するため、一般的に誤差が発生する。そこで行列 $\mathbf{V}$ と $\mathbf{T} \times \mathbf{X}$ の距離 $\mathbf{D}$ (式(5))を定義

表 1 擬音語と対応づけた特徴量

物理パラメータ	
物体 A	質量, 粘性
物体 B	質量, 体積, 速度

表 2 回帰モデル生成における収束時間

実験数	パラメータ数	収束時間[s]
10	5	0.997
20	5	1.265

し, これを最小化することを考える. 距離 D を以下の式(5)に示す.

$$D(V, TX) = \sum_i \sum_j d(v_{ij} - t_i^T x_j) \quad (5)$$

上式を最小化のために用いる距離尺度に, KL-divergence(一般化 Kullback-Leibler divergence)を用いた式(6)^{[17][18]}.

$$d_{KL}(v_{ij} - t_i^T x_j) = v_{ij} \log \frac{v_{ij}}{t_i^T x_j} - v_{ij} + t_i^T x_j \quad (6)$$

但し, 推定する際に用いた擬音語の中で一度も現れなかったアルファベットを含めて回帰モデルを生成すると, 収束しない可能性がある(文字列行列 $\mathbf{X}$ の中で, いずれかの列成分が全てゼロの場合). そこで, Multiplicative update rules を繰り返し適用する際に, 学習に用いた擬音語に存在しなかった文字のパラメータを $\mathbf{0}$ として固定し, 更新式から除外した. 収束に要した時間を表 2 に示す.

3.2 擬音語の最適化

本章はシミュレータやアニメーション中で決定する物理パラメータ $\mathbf{V}_{\text{ext}}$ を用いて, 擬音語を推定する手法を説明する. 具体的には前章で推定した擬音語の回帰モデル $\mathbf{T}$ に対し, 最適な文字列ベクトル $\mathbf{W}$ を算出し, アニメータが利用する擬音語リストの中から検索するというものである.

通常, 回帰モデル $\mathbf{T}$ を基にアニメータが用いる擬音語に物理的な印象を付与するのが一般的である. しかし, 学習で計算された回帰モデルは, アニメータが使用する特殊な文字 (例: 「@」など) にパラメータを決定することはできない. 例えば「L」「!」が回帰モデル $\mathbf{T}$ に存在しない場合, 「BAM」「BLAM」「BAM!!!」のパラメータが一致してしまうことや, 回帰モデル $\mathbf{T}$ に存在しない文字のみで構成される擬音語のパラメータがゼロとなってしまう, 擬音語検索ができない. 解決策として, 回帰モデル $\mathbf{T}$ に存在しなかった文字に一定のペナルティ関数を設定することが挙げられるが, 物理パラメータ $\mathbf{V}_{\text{ext}}$ の各成分は質量や粘性といった次元が大きく異なるため, 適切な値を決定することが困難である. さらに別のアニメータが使用する擬音語リストに

切り替える時や再学習する場合, すべての擬音語に物理パラメータを付与する必要があるため現実的ではない.

そこで擬音語の文字列を基に, 物理パラメータを決定する手順とは逆に, 物理パラメータ $\mathbf{V}_{\text{ext}}$ を基に文字列ベクトル $\mathbf{W}$ を決定する手法を提案する. 物理パラメータと異なり, 文字列ベクトル $\mathbf{W}$ は各成分の次元が一致しているため, 回帰モデルに存在しない文字に対して, 一律のペナルティ関数を設定することが可能となる. さらにアニメータが考える擬音語の文字列を直接扱うことができるため, 再学習の度に物理パラメータを付与する処理を省くことができる. 文字列は正值であることや, 擬音語の文字数(3文字~8文字)の制約を加え, 以下の評価関数(7)を最小にする文字列ベクトル $\mathbf{W}$ を決定した.

$$f(t) = |\bar{\mathbf{V}}_{\text{ext}} - \mathbf{T}\bar{\mathbf{W}}(t)|^2 \quad (7)$$

$$\begin{cases} w_i \geq 0 & (i \in R^D) \\ 8.0 \geq \sum_i w_i \geq 3.0 \end{cases}$$

上式の制約有り非線形問題を解くために SUMT と準ニュートン法を用いた.

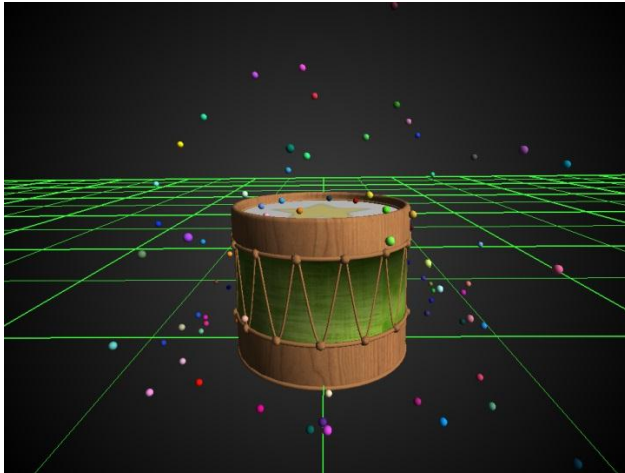
次に算出した文字列ベクトル $\mathbf{W}$ を基に最適な擬音語を決定する手法の手順を述べる. 文字列ベクトル $\mathbf{W}$ を基に擬音語として尤もらしい文字列を決定することも可能だが, 本稿の目的はアニメータが考える擬音語の中から最適な擬音語を決定することにある. そこで文字列ベクトル $\mathbf{W}$ とデータベース中に存在する擬音語の文字列ベクトル $\mathbf{D}$ との類似度を計算し, 類似度が最も高い擬音語を選択する手法を提案する. 文字列同士の類似度を計算するためにマハラノビス距離を用いた. ただし回帰モデルに存在しない文字には, ペナルティ関数 f を第二項に追加した. 類似度を計算する式(8)は以下に示す.

$$O = \min_i \left\{ \left| (\bar{\mathbf{W}} - \bar{\mathbf{D}}_i)^T \mathbf{S}^{-1} (\bar{\mathbf{W}} - \bar{\mathbf{D}}_i) \right| + \alpha \sum_j f(j) \right\} \quad (8)$$

$$f(j) = \begin{cases} 1.0 & \text{if } j \in D \\ 0.0 & \text{else} \end{cases}$$

$\mathbf{S}$ は共分散行列, α は重み係数である.

以上の手順で外部パラメータに対し, 最適な擬音語を決定した. またアニメータが映像の演出に応じて擬音語を修正する場面を想定し, 再学習機能を追加した. 再学習とは, アニメータの擬音語リストの編集ではなく, 回帰モデル $\mathbf{T}$ を改善することである. そこで学習に用いた k 種類の擬音語に対し, 各ユーザが編集した擬音語と物理パラメータを $(k+1)$ 番目の情報として式(2), (4)に加え, 回帰モデルを再度生成することで, 各ユーザの意思を反映した擬音語推定が可能になると考えられる.



(a) 入力アニメーション



(b) 擬音語付与結果

図5 生成結果の一例（ドラムアニメーション）

3.3 擬音語の描画位置推定

本章は自動推定された擬音語を描画する位置を決定する。複数のアニメ作品を検証し、擬音語の位置を衝突に対する反射方向とした。ただしアニメ作品と同様に、描画した擬音語同士が重ならないように制約を加える必要がある。解決策として、擬音語の初期値を反射方向に、複数の擬音語を描画する際には、以前に描画した擬音語の位置（3次元空間上の擬音語の分布）を基に、描画位置をずらすことが挙げられる。j番目の擬音語の位置 $\mathbf{p}_j$ を基に i 番目に描画する擬音語の位置 $\mathbf{p}_i$ を以下の評価関数 E_i (式(9))を最小化することで決定した。

$$E_i = \left| \bar{\mathbf{p}}_i - \sum_{j=0}^{j < i} \beta(t) \exp \left(- \text{dist}(\bar{\mathbf{p}}_i, \bar{\mathbf{p}}_j) / 2\sigma^2 \right) \right|^2 \quad (9)$$

$$\beta(t) = k_{font} t(t_{max} - t)$$

t_{max} は擬音語が表示できる最大時間、 σ は擬音語の文字サイズに応じたサポート半径に基づき決定する（図5）。

また3次元空間に描画する際、擬音語を読めるように視線方向に対しローカル軸を回転させる必要がある。そこでカメラの視線方向(Euler角)とカメラの位置座標を基に、擬音語を視線方向に対し垂直となる平面上に配置する。さらに擬音語の位置とカメラ座標の距離に応じてフォントサイズ、擬音語の描画時間に応じて文字の色を決定し、フォントの種類はインタラクティブに変更する機能を追加した。

4. 生成結果

以上の手順によって、OpenGL で実装したプロトタイプを図5,7~8に示す。プロトタイプは、物理シミュレーション上で必要となる物理パラメータをユーザが調整することで、擬音語をリアルタイムで推定及び付与することができ

るシステムとなっている。またアニメータが考える擬音語のリストの中から、自動で推定した擬音語に加え、類似度の高い擬音語を複数提示することで、各ユーザの好みに応じて編集することができる。さらに編集した結果に応じて学習データを更新することによって、各ユーザの好みを反映し、見ごたえのある擬音語アニメーションを生成することも可能である。その他の機能として自動推定した擬音語に加え、従来の漫画的な表現手法である効果線と衝撃マークを付与する機能を追加した。

5. 精度評価と実行速度

本システムによる擬音語推定の有用性を検証するために、学習の際に擬音語と対応付けした物理パラメータに着目する。事前学習の際に用いたアニメーションの物理パラメータを入力とし、データベースからユーザが選択した擬音語を検索できるかの実験を行った。その結果を表2に示す。その結果、ランダムに検索する場合と比較し、約5~9倍の精度で適切な擬音語を決定することが可能となった。これ

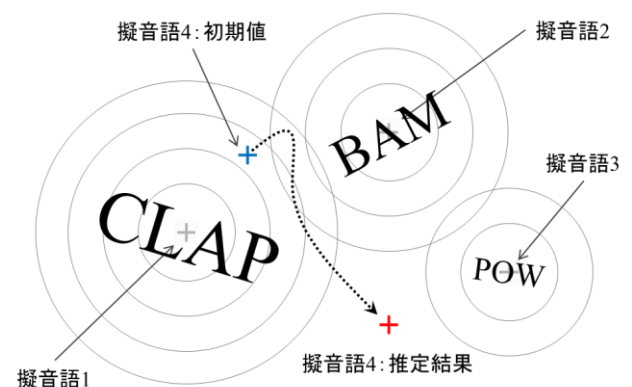


図6 擬音語の描画位置の概念図



(a) 入力アニメーション



(b) 擬音付与結果

図 7 生成結果の一例 (弾性体シミュレーション)

はアニメータが擬音語を、文字構成を基に決定しているという経験的な法則を定量化する可能性を証明したと言える。またユーザのインタラクティブな編集や再学習によって、精度が向上すると考えられる。

次に、本手法の実行速度についても考察する。実行環境は Intel(R) Core(TM) i7-3770 CPU @3.40Ghz Windows 7, 8.00GB 64bit である。頂点数の異なる 3D モデルでの実行速度を表 3, 4 に示す。従来の弾性体シミュレーションやキーフレームアニメーションと比較し、約 5[%]程度の処理負荷で擬音語を付与したアニメーションを実現した。この結果から、その他のアニメーションに適用した場合にも高速処理が可能であると考えられる。また本システムの簡単なインタビューを行い、システムについての簡単なフィードバックをもらった。意見として「見ごたえがある」「通常のアニメーション音語のフォントの最適化手法が欲しい」「擬音語の軌道を編集する機能が欲しい」などの意見もあった。今後はインターフェースとしての機能をより充実させ、フォントに関しても最適化する機能を検討したい。

6. まとめと今後の課題

本稿は、CG アニメーションの制作において必要となる物理的なパラメータを基に、擬音語を自動推定及び付与する手法を提案した。それにより、アニメータが感覚的・経験的に決定する擬音語を定量的に扱うことができるようになり、アニメ作品の制作において最も時間を要する箇所の効率化につながったと言える。また自動推定した擬音語に加え、類似度の高い擬音語候補を提示することで、よりユーザの好みに応じて、見ごたえのある擬音語アニメーションを生成することが可能となった。さらに従来のアニメーションに対し、約 5[%]程度の処理負荷で擬音語付与を実現した。

表 2 精度評価

基底擬音語数	Our Method[%]	Random[%]
10	50.0	10.0
20	45.0	5.0

表 3 処理速度(弾性体シミュレーション比較)

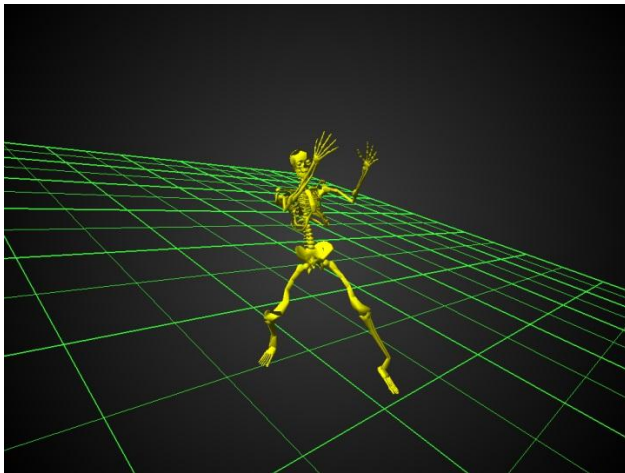
Model	Vertex No	Our Method[fps]	Shape Matching[fps]
Teapot	530	749.89	762.67
Rabbit	4138	185.47	199.82
The Bunny	16292	52.46	54.75
King Kong	23581	32.84	34.88

表 4 処理速度(キーフレームアニメーション比較)

Model	Vertex No	Our Method[fps]	SSD[fps]
Skeleton	4138	58.47	60.98

また擬音語を付与することによって、入力アニメーション(図 5(a), 7(a), 8(a))の静止画では判断が困難である、衝突現象が発生している箇所を一目で把握することが可能となった。擬音語は元来、漫画のような静止画や動きのないコマに、動きを生み出すための補助ツールとして用いられる。そこで CG アニメーションだけでなく、動画中の動きを解析し、静止画 1 枚に擬音語を描画する映像要約手法として応用できると考えられる^{[19][20]}。手順として動画中のキーフレームを検出し、事前学習の際に動画中の画像特徴量と擬音語の対応を取ることで実現すると考えられ、今後着手していきたい。

日常生活で知覚する音には、人の音声と音楽、環境音がある。これらを判定することで、音情報による環境理解



(a) 入力アニメーション



(b) 擬音付与結果

図 8 生成結果の一例 (キャラクターアニメーション)

や危険察知などへの研究が提案されている。具体的には、音源名でなく「音がどのような状況で発生しているか」という情報のシンボルとして、擬音語が注目されている。本手法は、主観的に対応付けを行った擬音語と物理パラメータに着目した手法であるため、特徴量の種類に制限されない。そこで擬音語と音声の対応付けることで、音声を基に擬音語を推定することも可能である。今後は音響情報と擬音語の関係を解析する予定である。

本手法は既存のアニメーションの動きを強調・補助するため手法であり、キャラクターのモーションやメッシュ変形自体を誇張する手法ではない。そこでアニメ的な動きに変換する手法^[21]と併用することでより、更なるアニメ制作の効率化を目指したい。

謝辞 本研究の一部は JST「OngaCREST プロジェクト」の支援を受けた。

参考文献

- 1) Johannes Schmid, Robert W. Sumner, Huw.Bowles, and Markus Gross. 2010. **Programmable Motion Effects**. ACM Transaction on Graphics (TOG) – Proceeding of ACM SIGGRAPH 2010, ACM New York, NY, USA, volume 29 Issue 4, Article No.57.
- 2) Maic Masuch, Stefan Schlechtweg, and Ronny Schulz. **Speedlines: depicting motion in motionless pictures**, Proceeding ACM SIGGRAPH'99 Conference Abstracts and Applications, page 277, New York, NY, USA, 1999.
- 3) Marc Nienhaus and Jurgen Dollner. **Depicting Dynamics using principles of visual art and narrations**. IEEE Computer Graphics and Application, Volume 11, Issue. 3, pp.40-51, 2005.
- 4) Daiki Umeda, Tomoaki Moriya, and Tokiichiro Takahashi. **Real-time Manga-Like Depiction Based on Interpretation of Bodily Movements by Using Kinect**, Proceeding SA'12 SIGGRAPH Asia 2012 Technical Briefs Article No.28, ACM New York, NY, USA, 2012.
- 5) Yoshinori Dobashi, Tsuyoshi Yamamoto, and Tomoyuki Nishita, **Real-time Rendering of Aerodynamic Sound Using Sound Textures based on Computational Fluid Dynamics**, ACM Trans on Graphics, vol.22, No.3 (Proc. SIGGRAPH2003), pp.732-740, 2003-7.
- 6) Jeffrey N Chadwick, Changxi Zheng, and Doug L. James, **Precomputed Acceleration noise for Improved Rigid Body Sound**, ACM Transactions on Graphics (SIGGRAPH2012), 2012.
- 7) Kazushi Ishihara, Yasushi Tsubota, and Hitoshi G. Okubo: **Automatic Transformation of Environmental Sounds into Sound-Imitation Words Based on Japanese Syllable Structure**, Proceedings of EUROSPEECH 2003, pp.3185-3188, 2003.
- 8) Jaap Blonk and Golan Levin, **Ursonography**, the Ars Electronica Festival, September 2005..
- 9) Dan B Goldman, Brian Curless, David Salesin and Steven M. Seitz. **Schematic Storyboarding for Video Visualization and Editing**. ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2006), Vol.25, No.3, pp.862-871, July 2006.
- 10) Takanori Komatsu, **Quantifying Japanese Onomatopoeias: Toward Augmenting Creative Activities with Onomatopoeias**, Proceedings of the 3rd Augmented Human International Conference Article No.15, ACM New York, NY, USA, 2012.
- 11) Joe Khatena, **Onomatopoeia and Images: Preliminary validity study of a test of Originality**. Perceptual and Motor Skills, vol 28(1), pp.335-338. 1969.
- 12) Yutaro Tomoto, Tsuyoshi Nakamura, Masayoshi Kanoh and Takanori Komatsu, **Visualization of Similarity Relationships by Onomatopoeia Thesaurus Map**, Proceedings of the 2010 IEEE International Conference on Fuzzy Systems (FUZZ), pp.1-6, 2010.
- 13) Yuya Ueda, Yuichiro Shimizu, and Maki Sakamoto, **System Construction Supporting Communication with Foreign Doctor Using Onomatopoeia Expressing Pains**,

- Soft Computing and Intelligent Systems(SCIS) and 13th International Symposium and Advanced Intelligent System(ISIS), pp.508-512, 2012.
- 14) David C. Burr., and John Ross., 2002. **Direct evidence that “speedlines” influence motion mechanisms.** Journal of Neuroscience 22, 19.
 - 15) Matthias .Muller, Bruno Heidelberger, Matthias Teschner, and Markus Gross. **Meshless Deformations based on Shape Matching.** ACM Transactions on Graphics (TOG) – Proceedings of ACM SIGGRAPH2005, New York, NY, USA, Volume 24 Issue 3, page 471-478.
 - 16) Stephen Chenney, Mark Pingel, Rob Iverson, Marcin Szymanski. **Simulating Cartoon Style Animation** , Proceeding NPAR’02 Proceeding of the 2nd International Symposium on Non-photo realistic animation and rendering, pp.133-138, ACM, New York, NY, USA, 2002.
 - 17) Daniel D. Lee and H.Sebastian Seung. **Learning the parts of objects by non-negative matrix factorization,** Nature 401, pp.788-791, 1999.
 - 18) Daniel D. Lee and H.Sebastian Seung. **Algorithms for Non-negative Matrix Factorization,** Neural Information Processing Systems Foundation (NIPS), pp.556-562, 2000.
 - 19) Myung Geol Choi, Seung-Tak Noh, Taku Komura, and Takeo Igarashi, 2013. **Dynamic Comics for Hierarchical Abstraction of 3D Animation Data.** Computer Graphics Forum (Proceedings of Pacific Graphics 2013).
 - 20) John.P.Collomosse, D.Rowntree, and Peter.M.Hall. **Rendering Cartoon-style Motion Cues in Post-production Video,** Graphical Models, Volume 67, Issue 6, pp.549-564, 2005.
 - 21) Tomohiko Mukai, Shigeru Kuriyama, **Geostatistical Motion Interpolation,** ACM Transactions on Graphics (SIGGRAPH 2005), 24(3), pp.1062-1070, 2005