Automatic Depiction of Onomatopoeia in Animation Considering Physical Phenomena

Tsukasa Fukusato*
Waseda University / JST

Shigeo Morishima†
Waseda Research Institute for Science and Engineering / JST

Figure 1: *Comparison of input elastic animation (left) with the proposed onomatopoeia animation method, a mechanism for depicting onomatopoeia in computer-generated animation (right). The proposed onomatopoeia depiction method enables enhancement of character movement, and strength of collision impact by considering physical phenomena (the 3D models "Big Buck Bunny" ©Blender Foundation).*

Abstract

This paper presents a method that enables the estimation and depiction of onomatopoeia in computer-generated animation based on physical parameters. Onomatopoeia is used to enhance physical characteristics and movement, and enables users to understand animation more intuitively. We experiment with onomatopoeia depiction in scenes within the animation process. To quantify onomatopoeia, we employ Komatsu's [2012] assumption, i.e., *onomatopoeia can be expressed by n-dimensional vector*. We also propose phonetic symbol vectors based on the correspondence of phonetic symbols to the impressions of onomatopoeia using a questionnaire-based investigation. Furthermore, we verify the positioning of onomatopoeia in animated scenes. The algorithms directly combine phonetic symbols to estimate optimum onomatopoeia. They use a view-dependent Gaussian function to display onomatopoeias in animated scenes. Our method successfully recommends optimum onomatopoeias using only physical parameters, so that even amateur animators can easily create onomatopoeia animation.

CR Categories: I.3.3 [Computer Graphics]: Three-Dimensional Graphics and Realism—; I.3.6 [Computer Graphics]: Methodology and Techniques—;

Keywords: quantification of onomatopoeia, onomatopoeia vector, phonetic symbol vectors, view-dependent onomatopoeia depiction

*e-mail: tsukasa@moegi.waseda.jp
†e-mail: shigeo@waseda.jp

Motion in Games 2014, November 06 – 08, 2014, Los Angeles, California.

http://dx.doi.org/10.1145/2668064.2668096

1 Introduction

Japanese cartoon animation (or anime) have been attracting worldwide attention due to their artistry and unique story-lines. To present a world view in anime, anime-like physical techniques, such as sound symbolic and mimetic words are used. Sound symbolism is a term used in semiotics and linguistics to refer to a direct association between the form and meaning of language; sounds reflect the properties of the external world. In linguistics, sound-symbolic words can be classified as follows: phonomime, onomatopoeia, and psychomimes. Onomatopoeias mimic actual sounds, phenomines depict nonauditory senses, and psychomimes depict psychological states or bodily feelings.

Onomatopoeias are often used to enhance the physical characteristics and motion of a character, and enable users to understand unrealistic anime contents intuitively. Recently, onomatopoeias have attracted attention in print media and literary work, such as picture books. However, the automatic selection of optimum onomatopoeia for a scene has two problems. First, sound-symbolic words are not only imitative of sound but also cover a much wider range of meaning. Second, selecting onomatopoeias based on sounds during anime creation is very difficult because many production studios use after-recording which is a general method for recording speech to accompany the finished visuals of anime films. Thus, animators must select onomatopoeias only based on character reference and motion. To maintain quality, significant manual labor is required even though computer-generated (CG) techniques reduce labor intensity. Moreover, many amateur animators cannot assign optimum onomatopoeias to animation.

Our goal is to estimate and depict onomatopoeias based on static and dynamic physical parameters that are computed using CG animation. The main idea of our approach is to formulate empirical knowledge of sound symbolism. The overall process of the proposed method is as follows:

Quantification of phonetic symbol

1. Viewing simple CG animations, e.g., ball bouncing, and selecting suitable onomatopoeia. Consequently, onomatopoeias are associated with computed physical parameters according to simple animation.

2. Factorizing physical parameters computed in an animation simulation process into phonetic symbols, and creating a phonetic symbols vector matrix.

Onomatopoeia estimation

3. Inputting external parameters computed in an arbitrary animation and estimating onomatopoeia using phonetic symbol vectors automatically.

4. Depicting multiple onomatopoeias in three-dimensional (3D) space under the constraint that they do not overlap.

The proposed method enables automatic estimation and recommendation of optimum onomatopoeias for any CG animation. Furthermore, we enable the creation of animation with more suitable onomatopoeia through an editing and relearning function. The proposed method makes the manual labor processes of creating anime-like animation more efficient. In addition, it is expected that the proposed method can enable a summarization method for depicted onomatopoeia.

The remainder of this paper is organized as follows. Related work is reviewed in Section 2, and we discuss the main ideas underlying the algorithms used in the proposed method in Section 3. In Section 4, we describe an implementation detail of our prototype system. Section 5 presents results, and we conclude the paper and discuss limitations and future work in Section 6.

2 Related Work

2.1 Depicting Cartoon Effects

Recently, research into generating anime-like effects has been proposed. Schmid's [2010] methods generate anime-like speed lines, motion blur, and dynamic glyphs by inputting keyframe animation. However, this method requires rebuilding 3D model-based time series information. In addition, significant analysis time is required. Umeda [2012] has proposed a system to depict anime-like effects based on simple image processing using joint data acquired by a Microsoft Kinect sensor. This method depicts regular speed lines and fonts associated with human motion; however, that study did not focus on anime characters or onomatopoeias.

Dobashi [2005] proposed a real-time physics-based sound simulation method to depict wind noise using fluid simulation and sound textures. However, it focused only on an object's shape and did not reflect physical characteristics which are required to select onomatopoeia. Chadwick [2012] proposed a method to generate acceleration noise for a rigid body. This method enables simulation of the sounds of rigid collision that reflect physical characteristics; however, sounds were not associated with most onomatopoeia in anime films.

Specific to caption, Hong [2010] proposed to visualize caption using image or sound features, and to color fonts considering time series information. With this method, it is possible to accentuate animation and video with caption easily; however they use a sound clustering method, (i.e., a multi-clustering method) to color subtitle. Unfortunately, it is difficult to classify sounds into various onomatopoeias.

Table 1: *An example of Sound-Symbolism.*

Form	Motivated	Example of symbolic words
[a]	large	large vast grand
[i]	small and thin	little petit piccolo
[u]	dark	blue glum
[b]	dull impact	bang bash bump
[j]	up and down movement	jump jangle jig
[g]	cracking	bang, bong
[cr]	noisy impact	crash crack crunch
[sl]	smoothly wet	slime slop slip
[sn]	quick separation or movement	snap snatch

2.2 Analyzing Onomatopoeia

Komatsu [2012] investigated the impression vectors of an alphabet composing onomatopoeia (sharpness, softness, dynamic, and largeness) by subjective experimentation. The most general Japanese onomatopoeias consist of the repetition of two syllables (e.g., MOKO-MOKO and PIYO-PIYO, i.e., the so-called XYXY-type). Thus, Komatsu assumed that XYXY-type onomatopoeias can be quantified by the following equation.

$$\vec{I_i} = a_i\vec{C}_{iX} + b_i\vec{V}_{iX} + c_i\vec{C}_{iY} + d_i\vec{V}_{iY} \quad (1)$$

Here $\vec{I_i}$ is the ith dimension of onomatopoeia expression vector. a_i, b_i, c_i, and d_i are weight coefficients, and $\vec{C_{iX}}$ and $\vec{V_{iX}}$ are the i-th respectively the i-th dimension values of X's consonant and vowel and $\vec{C_{iY}}$ and $\vec{V_{iY}}$ are the ith dimensional values of Y's consonant and vowel vectors. However, Komatsu determined alphabets that are caused by the impression of onomatopoeias empirically. Furthermore, in a questionnaire-based investigation to comprehend the multifaceted impression of alphabets, it is difficult for participants to fill in there 43 pairs of adjectives for alphabets.

In linguistic research, semantic similarities, such as the pronominal words *"the, this, thus, and there"* and the consonant sounds of negation words *"no, not, never, and neither"* are observed in English. This linguistic phenomenon is referred to as sound symbolism. Crystal [2003] defined sound symbolism as a term used in semiotics and linguistics that refers to a direct association between the form and meaning of language. The sounds used reflect the properties of the external world, as is the case with onomatopoeia and other forms of synesthesia. In addition, Lyons [1977] studies of phonesthesia and Jespersen's [1922] model, for example, propose that [fl-] [sl-] [gl-] are related to *"sound and view,"* and [i] and [u] means *"light"* and *"dark"* respectively. Bloomfield [1933] showed that [fl-] is *"moving light (flash, flame)"* and *"movement in air (fly, flap),"* and [sl-] is *"smoothly wet (slime, slush, slop)."*

Table 1 shows the examples of vowels and consonants related to motivation and iconicity. Kohler [1927/1947] suggests that voiceless plosives [p, t, k] indicates *"linear and angular shape,"* while resonance [m, n, r, l] indicates *"roundish shape."* Ullman [1962] found that symbolic words, such as onomatopoeic words, have phonetic motivation and iconicity, and Lyons [1977] stated that onomatopoeia represent non-arbitrary relationship. Sound symbolism studies have confirmed the universality and iconicity of sound-symbolism. Furthermore, we confirm that the direct association between the form and meaning of language is the most important factor in linguistics.

In anime and comics, realistic physical characteristics and smooth motion are rare; thus, it is difficult to understand object motion

Figure 2: *Example of onomatopoeia in anime and comic. Top: a scene from Super Smash Bros ©Nintendo Co., Ltd). Bottom: a comic frame from Klonoa ©BANDAI NAMCO Games Inc.).*

and unusual stories. Therefore, onomatopoeias help users review anime information because onomatopoeias reflect the properties of the external world. However, to create onomatopoeia animation, animator must select onomatopoeia without sound features because CG animation is created only based on some physical parameters. Moreover, there is a large number of onomatopoeia for each unique anime film; thus we cannot assign optimum onomatopoeia.

We propose a method to quantify the impression vectors of onomatopoeia based on sound symbolism, to estimate onomatopoeia using only physical parameters, and to depict onomatopoeia in a scene. The proposed method is of great value, and can suggest onomatopoeia automatically and create anime-like CG animation. In addition, the algorithm used in the proposed method can learn the physical parameters associated with onomatopoeia interactively.

3 Onomatopoeia Depiction Method Principles

Our quantification of onomatopoeia framework is mainly inspired by Komatsu's method. First of all, we assumed that phonetic symbols can be expressed by n-dimensional vector based on empirical knowledge of sound symbolism, e.g., *"g, z, d, and b are muddy sounds suggesting big heavy, or dirty."* Any instance of onomatopoeia can then be expressed of expression vector (so called *"Onomatopoeia Vector"*) by linearly coupling the phonetic symbolism vectors. Therefore, to realize the quantification method for onomatopoeias, two steps must be achieved 1) determine the numbers and types of the dimensions of phonetic symbol vectors, and 2) determine the value for each dimension of the vectors.

Table 2: *International Phonetic Alphabet classification of vowels and consonants.*

Name	Class	Phonetic symbol
close vowel	0	/ɪ/ /u/
close-mid vowel	1	/e/ /o/
open-mid vowel	2	/æ/ /ɜ/ /ʌ/
open vowel	3	/ɑ/ /a/
voiceless-plosive	4	/p/ /t/ /k/ /ʧ/
voiced plosive	5	/b/ /d/ /ʤ/ /g/
implosive	6	/ɓ/ /ɗ/ /ɠ/
voiceless fricative	7	/f/ /θ/ /s/ /ʃ/
voiced fricative	8	/v/ /ð/ /z/ /ʒ/
nasal	9	/m/ /n/ /ŋ/
lateral	10	/l/ /j/
approximant	11	/r/ /w/ /h/

Specially, we focus on *"onomatopoeia of two-body collision animations"* because collision scenes can benefit from various onomatopoeias in anime. Figure 2 shows an example of onomatopoeia in anime and comic.

3.1 Phonetic Symbol Vectors and String Vectors of Onomatopoeia

To determine the numbers and types of dimensions of phonetic vectors, we used the synesthetic knowledge of sound symbolism of eleven International Phonetic Alphabet (IPA) classifications, for example plosive consonants suggest *"un-expected phenomena,"* nasal consonants can evoke *"acoustic echo,"* and close vowels infer *"small."* Table 2 shows that IPA classification (the ordinal number of IPA classification). In linguistics, synesthetic sound symbolism is the use of sound to symbolize object size, shape, and speed [Jespersen, 1922]. Then, we assumed that the types of the dimension of phonetic vectors are *"mass"*, *"volume"*, *"acceleration"* and *"viscosity."*

However, we focus on *"onomatopoeias of two-body collision animations"*, as shown in Figure 3. That is, we must consider impression of two objects. We selected the six factors, mass(object A), volume(object A), acceleration(object A), mass(object B), viscosity(object B), and acceleration(object B), as shown in Table 3. Therefore six factors became the six dimensions of the phonetic vectors ($n = 6$) and onomatopoeia vectors can be expressed as *Vector* = $(mass_A,\ volume_A,\ acceleration_A,\ mass_B,\ viscosity_B,\ acceleration_B)$, i.e., $\vec{V_i}$ is expressed as follows.

$$\vec{V_i} = (v_0, v_1, \cdots, v_n)^{\mathrm{T}} \tag{2}$$

Similarly, the phonetic symbol vector $\vec{T_j}$ is expressed as follows.

$$\vec{T_j} = (t_0, t_1, \cdots, t_n)^{\mathrm{T}} \tag{3}$$

j is the class number of IPA classification(as shown in Table 2). Figure 4 shows that onomatopoeia, *"BANG,"* can be quantified by the linearly coupling the sound symbolism vectors.

The phonetic symbol rules defined by the $n \times D$ matrix T. D is the number of IPA classification (in this paper, $D = 11$). If we rewrite them in matrix form treating the vectors as rows, we get matrix T where

$$T = (\vec{T_0}, \vec{T_1}, \cdots, \vec{T_D}) \tag{4}$$

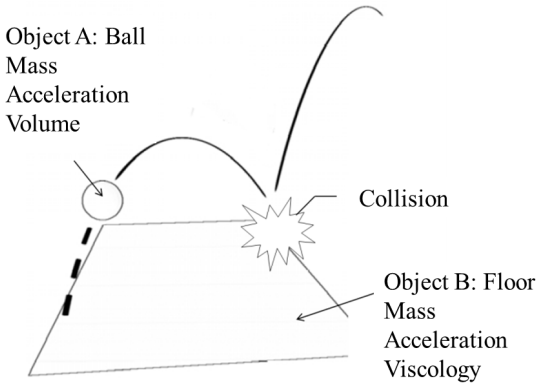


Figure 3: *Example of two-body collision animation in questionnaire-based investigation (Section 3.1). After viewing the animations, participants answered which onomatopoeia they want to depict in the scene (Section 3.2).*

Table 3: *Parameters associated with onomatopoeia.*

	Physical parameters
Object A	mass, volume, acceleration
Object B	mass, viscosity, acceleration

To resolve the matrix T, we defined the string vector $\vec{X}_i$ by counting types phonetic symbol of onomatopoeia. The column vector components are a number of stored phonetic symbols based on the ordinal number of IPA classifications (as shown in *"class number"* of Table 2). The string vector is expressed as follows.

$$\vec{X}_i = (x_0, x_1, \cdots, x_D)^{\mathrm{T}} \tag{5}$$

Figure 5 shows an example of string vector, *"BANG."*

We suppose the quantification of phonetic matrix T is based only on one onomatopoeia. $\vec{V}_i$ is computed by linearly coupling the phonetic symbol vectors in Eqs(4) and (5), i.e.,

$$\vec{V}_i = T \cdot \vec{X}_i \tag{6}$$

3.2 Quantification of Phonetic Symbol

To determine the value for each dimension of phonetic symbol vectors, we performed a questionnaire-based investigation to comprehend the correspondence the impressions of onomatopoeia. Specifically, some CG animations (Figure 3: two-body collision animations) were prepared. Participants answered the impression value on a 10-point Likert scale to physical parameters of CG animation, (the value for each dimension of Eq(2)), such as *"large - small"* (volume of object A). Furthermore, participants were asked *"which onomatopoeia do you want to select for the animation?"* Two-body collision animations are elastic, rigid and character animations. In this investigation, we get the impression values of physical parameters computed CG animation and the correspondence of string vectors of onomatopoeia Eq(5) to onomatopoeia vectors Eq(2) determined by impression values. In this paper, the number of participants are five, and the number of investigation k are twenty three ($k = 23$). It took about 20 minutes to complete this investigation.

In Section 3.1, for each pair of onomatopoeias data (string vectors $\vec{X}_i$ and onomatopoeia vectors $\vec{V}_i$), we compute phonetic symbol

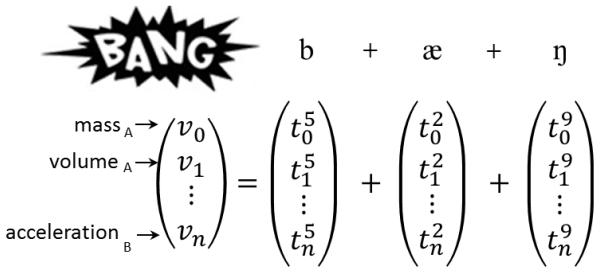


Figure 4: *Example of onomatopoeia vector $\vec{V}$, BANG, where is expressed by linearly coupling phonetic symbol vectors, [b], [a], and [ŋ].*

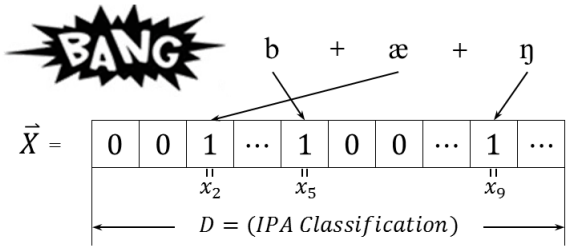


Figure 5: *The components of string vector $\vec{X}$ are a number of stored phonetic symbols based on the ordinal number of IPA classification, and have positive values.*

matrix T using Eq(6). However, in this step, we consider k onomatopoeias vectors rather than a single onomatopoeia.

In linguistics, since the phonetic symbols of onomatopoeia have synesthetic sound symbolism, a quantification of phonetic symbol vectors is conforming with all the individual ideal phonetic symbol values of onomatopoeia (in general). Therefore, we expand Eqs (4), (5), and (6) (in Section 3.1), and define X and V to be the matrix form treating the vectors (k onomatopoeia vectors and string vectors) as columns, we get X and V where

$$X = (\vec{X}_0, \vec{X}_1, \cdots, \vec{X}_k) \tag{7}$$

$$V = (\vec{V}_0, \vec{V}_1, \cdots, \vec{V}_k) \tag{8}$$

A closed form expression for Eq(6) is given by

$$V \approx T \cdot X \tag{9}$$

In order to compute the phonetic symbol matrix T, we rewrite Eq(9) for the divergence F between the onomatopoeia vectors V and the phonetic symbol matrix T and string vectors X ($= TX$) in Equation (10).

$$F(T) = \|V - TX\|^2 \tag{10}$$

The divergence functions are the Frobenius norm. We identify phonetic symbol matrix T by minimizing Eq(10). In addition, we assumed that the matrix T are non-negative matrix because the matrix V and X have no negative elements. That is to say, we assumed that Eq(10) has the constraint that the matrix T, V and X have no negative elements. In order to minimize the divergence $F(T)$, we used

Table 4: *Convergence time to compute phonetic symbol matrix T.*

The number of investigations (k)	The number of dimension of phonetic vectors (n)	Convergence time (s)
10	6	0.997
23	6	1.265

the multiplicative update rules, usually minimizing the divergence. The update rule is expressed as follow.

$$T_{kj} \leftarrow \frac{\sum_i V_{ki} X_{ji}}{\sum_i (\vec{T_k}^{\mathrm{T}} \cdot \vec{X_i}) X_{ji}} \tag{11}$$

As the result, we can confirm that the vectors for voiceless plosives [p, t, k] and resonances [m, n, r, l] have more intense values than vowels. Table 4 shows the convergence time required to compute the phonetic symbol vector matrix T.

3.3 Estimating Onomatopoeia

In this section, we describe a method to recommend optimum onomatopoeia in database for external onomatopoeia vector $\vec{V_{ext}}$ determined by physical parameters of an arbitrary CG animation. The onomatopoeia database is word list of onomatopoeias which consists of 100 collision-related onomatopoeias selected from anime films that were among the top 10 films by annual DVD and comic book sales. Firstly, we automatically compute the optimum string vector $\vec{\omega}$, i.e., Eq(5), using an external onomatopoeia vector $\vec{V_{ext}}$ and the phonetic symbol matrix T, i.e., Eq(4). Our method enables us to compute any onomatopoeia vectors. However, if we compute many onomatopoeia vectors in database, time required for computing onomatopoeia vectors of database is directly proportional to the number of words in the database. That is to say, it is difficult to add an editing and relearning onomatopoeia vectors in database.

Furthermore, for example if there is not implosive sound of onomatopoeia in the questionnaire-based investigation (in Section 3.2), we cannot compute implosive sound vector $\vec{T_6}$ in phonetic symbol matrix T. We cannot compute onomatopoeia vector which include implosive sound. Therefore, it is necessary to add a constraint to non-learning phonetic symbols in the database, e.g., implosive sound. However, it is difficult for us to set appropriate values for the constraint because the value for each dimension of onomatopoeia vector V has a different dimension, e.g., *"mass and viscosity."*

Therefore, we proposed a method to estimate the string vector $\vec{\omega}$ based on external parameters $\vec{V_{ext}}$ rather than computing the onomatopoeia vector of any instance of onomatopoeia. Each component of the string vector $\vec{\omega}$ has the same dimensionality in Eq(5), i.e., it is possible to recommend an onomatopoeia using the string vector $\vec{\omega}$ and the constraint penalty function of the phonetic symbols. Furthermore, we need not compute all onomatopoeia vectors based on the phonetic matrix T. Instead, we can compute string vectors (5) in the database. Then, we solve the following constrained optimization problems for the string vector $\vec{\omega}$.

$$f(\vec{\omega}) = |\vec{V_{ext}} - T\vec{\omega}|^2 \tag{12}$$

$$\begin{cases} \omega_i \geq 0 & (i \in R^D) \\ 8.0 \geq |\vec{\omega}| \geq 3.0 \end{cases}$$

The following constraints apply: all string vector components are positive values and the number of onomatopoeia characters (the norm of string vector) is more than 3.0 and less than 8.0. We use the sequential unconstrained minimization technique and a quasi-Newton method to solve the optimization problem.

Although we can create new onomatopoeias based on string vector $\vec{\omega}$ and the combination rules for phonetic symbols, this is out of scope. The purpose of this study is to quantify and recommend optimum onomatopoeia for inclusion in an animators' onomatopoeia database. Therefore, we focus on selecting optimum onomatopoeia from a database, i.e., the retrieval of onomatopoeia composed of æ and ŋ. This process looks like *"Bag-of-features"* [Csurka et al, 2004]. To recommend optimum onomatopoeia, we calculated the degree of similarity between $\vec{\omega}$ and the string vector of onomatopoeia $\vec{d}$ in the database. We can present highly similar onomatopoeia in the database to string vector $\vec{\omega}$. To calculate similarity, we defined cost function $O(\vec{d})$ based on a binomial formula of Mahalanobis distance and a penalty function f. The cost function is expressed as follows:

$$O(\vec{d}) = |(\vec{\omega} - \vec{d})^{\mathrm{T}} S^{-1} (\vec{\omega} - \vec{d})| + \alpha \sum_j f(j) \tag{13}$$

$$f(j) = \begin{cases} 1.0 & if \quad d_j = 0, \omega_j > 0 \\ 0.0 & else \end{cases}$$

where matrix S is a covariance matrix based on string vector of onomatopoeia in the database and α is empirically set to 0.8.

Moreover, we added an editing function for relearning phonetic symbol matrix T. By adding $(k+1)$th editing data to Eqs(7) and (8), it is possible to obtain more suitable optimum onomatopoeia.

3.4 View-Dependent Onomatopoeia Depiction

In this section, we describe a method for depicting onomatopoeia in a CG scene. When onomatopoeias are depicted, it is essential that they should be rotated on a vertical plane relative to the viewing vector. Consequently, we rotate onomatopoeias based on the quaternion of the viewing vector. In addition, we add functions to compute the size of onomatopoeia, the render time by the force of object collision, and the editing font style interactively.

Some expressive font animation approaches that include controlled rapid rhythmic motions, changes of font size, and rotation have been studied. However, in many previous methods, it is necessary to input parameter values and waveform behaviors manually [Lewis et al. 1999][Lee et al. 2002]. Furthermore, it is not possible to apply these methods to all font behaviors in two-dimensional CG animation. Therefore, we considered a system to estimate onomatopoeia positions in 3D space. First, we analyzed anime films to determine default onomatopoeia positions. From this analysis, we assumed that the ith onomatopoeia default position $\vec{p_i}$ reflects the collision direction. Our primary consideration was that onomatopoeias should not overlap. Moreover, onomatopoeias are rotated based on the viewing vector; thus, viewing direction must be considered. We set a default and shift the ith onomatopoeia position $\vec{p_i}$ based on the view-dependent distribution of onomatopoeias from 0th to $(i-1)$th words. Therefore, we assumed that the distribution of onomatopoeia is a sum of Gaussian and determine onomatopoeia position $\vec{p_i}$ by estimating a small distribution. To minimize the onomatopoeia value, we defined the cost function $E(\vec{p_i})$ based on the distribution from the 0th to $(i-1)$th onomatopoeia and the viewing vector. The cost function is expressed as follows:

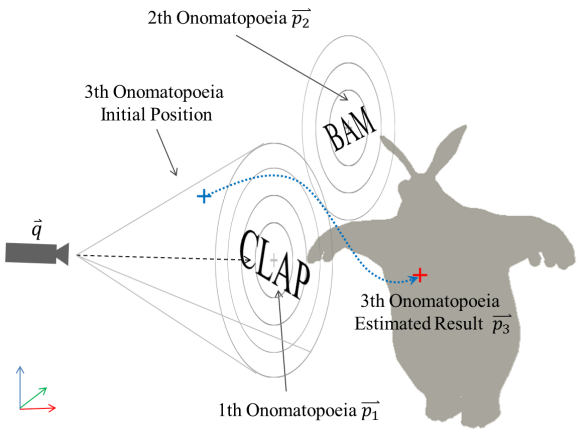


Figure 6: *Estimating onomatopoeia position $\vec{p_i}$ for depiction in a scene. When inputting 3th onomatopoeia, we optimize position $\vec{p_3}$ using a view-dependent Gaussian based on the 1&2th onomatopoeia positions.*

$$
\begin{aligned}
E(\vec{p_i}) &= \sum_{j=1}^{j<i} \beta(t) \exp\Big(\frac{\log w \cdot |\vec{h_{ij}}|^2}{|\hat{s_{ij}}|^2}\Big) \qquad (14)\\
\hat{s_{ij}} &= \Big(2.0 - \frac{|\vec{h_{ij}} - \vec{q}|}{|\vec{p_j} - \vec{q}|}\Big) \cdot s_j \\
\vec{h_{ij}} &= \vec{r_j} \cdot \frac{(\vec{r_j} \cdot \vec{r_i})}{|\vec{r_j}|^2} - \vec{r_i} \\
\vec{r_i} &= \vec{q} - \vec{p_i} \\
\beta(t) &= \begin{cases} k_f \cdot t(t_{max} - t) & t \le t_{max} \\ 0.0 & else \end{cases}
\end{aligned}
$$

where t_{max} is the maximum display time for an onomatopoeia, k_f is the constraint value, and s is the font radius computed according to font size and number of the characters. w is the distribution value of the font boundary, which is empirically set to 0.8. A quasi-Newton method is used to solve the ith position $\vec{p_i}$. Figure 6 illustrates the procedure for estimating onomatopoeia position $\vec{p_i}$ for depiction in a scene.

As a result of our attempt, depiction of onomatopoeias occasionally overlapped. It is assumed that onomatopoeia position can have been fallen into a local optimum solution in a gradient method. In future, it is essential to perform stable optimization without a local solution.

4 Implementation

Our prototype system is written using openFrameworks, an open source C++ toolkit. With our system, users can control physical parameters to simulate CG animation and depict optimum onomatopoeia automatically. Moreover, we can present highly similar onomatopoeia in the database to the string vector $\vec{\omega}$, and edit onomatopoeia according to user preference. It is possible to update phonetic symbols matrix T. In addition, we have added anime-like effects, such as *Speed-lines* and *Impact Mark* using Catmull-Rom spline curves. Furthermore, to select an optimum font for onomatopoeia automatically, we calculated the font similarity between optimum onomatopoeia and the pairs of onomatopoeia data (as shown in Section 3.2) and determined the font type. We use Levenshtein distance [Marzal et al, 1993] to calculate font similarity.

Table 5: *Onomatopoeia estimation results.*

Number of onomatopoeia	Our method (%)
23	47.8

5 Results and Discussion

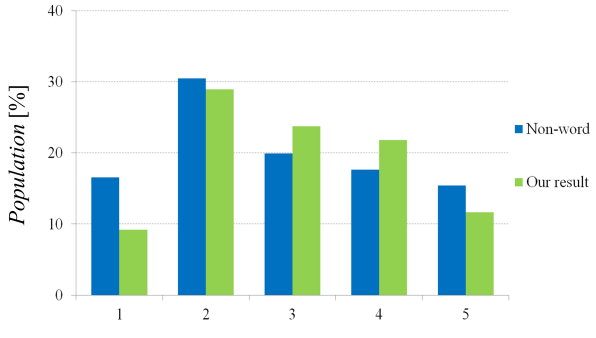
In this section, we discuss examples of animations depicting onomatopoeia using our proposed method. Our results are presented in Figures 1, 8 and 9. The figures indicate that onomatopoeias support the understanding of physical characteristics and character motion. Onomatopoeias also enhance anime-like effects, i.e., object motion and collision force.

We have verified the effectiveness of the proposed method by accuracy evaluation based on the correspondence between onomatopoeia string vectors $\vec{X}$ and onomatopoeia vectors $\vec{V_i}$ in the questionnaire-based investigation (as shown in Section 3.2). This closed test was performed to calculate the classification accuracy of k onomatopoeias by our method. In the experiments, the number of onomatopoeias string vectors corresponding to onomatopoeia vectors is twenty-three ($k = 23$). Table 5 shows the result of classification estimation for onomatopoeias using the proposed method. The results were obtained using an empirical rule: *any instance of onomatopoeia can be expressed as an expression vector by linearly coupling the phonetic symbol vectors.* Thus, it is possible to recommend onomatopoeia using phonetic symbols. In addition, relearning and editing phonetic symbol matrix T can interactively improve the accuracy of onomatopoeia classification. If we apply our onomatopoeia depiction system to various CG animations, for example dance animation, it is necessary to determine the number and types of dimensions of phonetic vectors for these animations. In future, we intend to investigate optimum dimensions of onomatopoeia vectors for various types of CG animation.

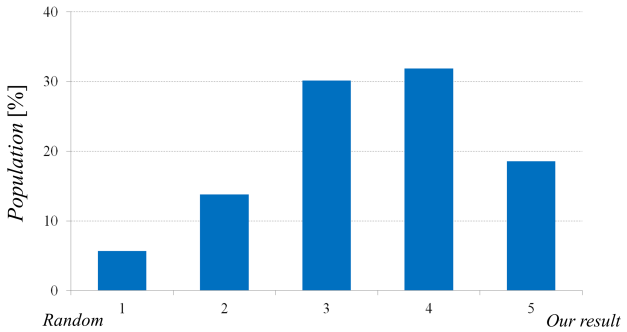
5.1 User Study

We also performed an experiment to assess our method through a user study. We recruited over 500 Japanese people through crowdsourcing. The participants were asked ten questions. Q1: *"I can feel the strength of collision and sound volume"* in a CG animation (without onomatopoeia) and our proposed method. Q2: *"I can feel the strength of collision and sound volume in an image."* Q3: *"It is easy to watch the animation."* Q4 - Q9: *"which of onomatopoeia animations is the optimum, our method or randomly chosen onomatopoeias."* Q10: *"which animation is more visually noisy in two animations, one animation using view-dependent onomatopoeia depiction of our proposed method or the other animation with screen coordinates based on projection matrix."* The answers were scored on a 5-point Likert Scale (AB scale). A high score of Q1 - Q3 means that the participants can understand the collision information and that the animation with onomatopoeia is more enjoyable than that without onomatopoeias. A high score of Q4 - Q10 indicates that our proposed method is effective.

Figure 7(a) shows the average distribution histograms of Q1 - Q3 (x-axis is 5-point Likert Scale and y-axis is the answer prevalence of all participants, so called *"population"*). As a result, depicting onomatopoeias is effective for gaining the dynamism of collision and motion and making animation enjoyable compared with normal animation. The mean score of Q1 - Q3 is 2.91 and the sample deviation of them is 1.09.



(a) *Score prevalence of all participants of our result and non-onomatopoeia animation(Q1-Q3).*



(b) *Score prevalence of all participants compared our result with randomly chosen onomatopoeias (Q4-Q10).*

Figure 7: *the score prevalence of all participants for each question (5-point Likert Scale and AB Scale).*

Table 6: *Question items.*

No	Question
1	I can feel the strength of collision and sound volume in a CG animation (without onomatopoeia/our result)
2	I can feel the strength of collisions and sound volume in an image.
3	It is easy to watch the animation. (without onomatopoeia/our result)
4 - 5	Which onomatopoeia is optimum in rigid animation? (AB scale)
6 - 7	Which onomatopoeia is optimum in character animation? (AB scale)
8 - 9	Which onomatopoeia is optimum in elastic animation? (AB scale)
10	Which onomatopoeia animations are posted in prominent place? (constraint position/our result) (AB scale)

Moreover, Figure 7(b) shows the average distribution histograms of Q4 - Q10 (x-axis is 5-point Likert Scale and y-axis is the answer prevalence of participants, *"population"*). The mean score of Q4 - Q10 is 3.55 and the sample deviation of them is 1.09. The result shows that the majority of participants thought that the proposed method is effective for gaining the anime-like information of collision force and object motion because onomatopoeias clearly reflect the properties of scenes.

Our prototype system was also evaluated by users who provided individual feedback: *"depicting onomatopoeias in the scene is more impressive than a normal CG animation,"* and *"I want a function of editing onomatopoeia positions."* In future, we plan to include position functions of onomatopoeia in the user interface to create richer animations.

5.2 Processing Speed

We verified the processing speed of the proposed method. A 64-bit Windows PC (Intel®Core^{TM} i7-3770 CPU@3.40GHz 8GB RAM; NVIDIA GeForce GT 620M 1 GB) was used. A comparison of our results with elastic simulation without onomatopoeia for different vertex models is shown in Table 7. Table 8, on the other hand, shows a comparison of our result with key-frame animation (skinning character animation) without onomatopoeias. The results show that the processing speed is approximately 5% compared with elastic simulation and keyframe animation.

Table 7: *Processing speed (Comparison of our results with elastic Simulation).*

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

Table 8: *Processing speed (Comparison of our results with skinning animation).*

Model	Vertex no	Our method (fps)	Skinning animation (fps)
Skeleton	4138	58.47	60.98

6 Conclusions and future work

We have presented the method to estimate and depict onomatopoeia based on physical parameters and sound symbolism in animated scenes. The prototype system enables the presentation of optimum onomatopoeia from the onomatopoeia database. The presented onomatopoeia can be edited according to user preferences. We proposed phonetic symbol vectors using a known empirical onomatopoeia rule, i.e., sound symbolism.

In future, we plan to increase the types of dimensions of onomatopoeia vector and focus on different aspects of displaying onomatopoeia, such as brightness and color. Since, we recognize that interactive font editing and positioning functions for onomatopoeia will help users to create richer animations more efficiently, we intend to address these functions. Such functions are applicable to a wide range of situations in anime production. We believe our technique can help animators reduce time, labor, and cost.

The proposed system enables the visualization associated with the physical characteristics of objects as well as characters or object motion, e.g., collision frequency. Onomatopoeia is a supporting tool that can enhance character motion. Moreover, the central idea of our approach is to associate physical parameters with onomatopoeia; therefore, it is assumed that our system could also be applied to comic-style video summarization systems to help users find and review required information based on image features. We intend to apply the proposed approach to live-action films and to generate comic-style video summarization using onomatopoeia.

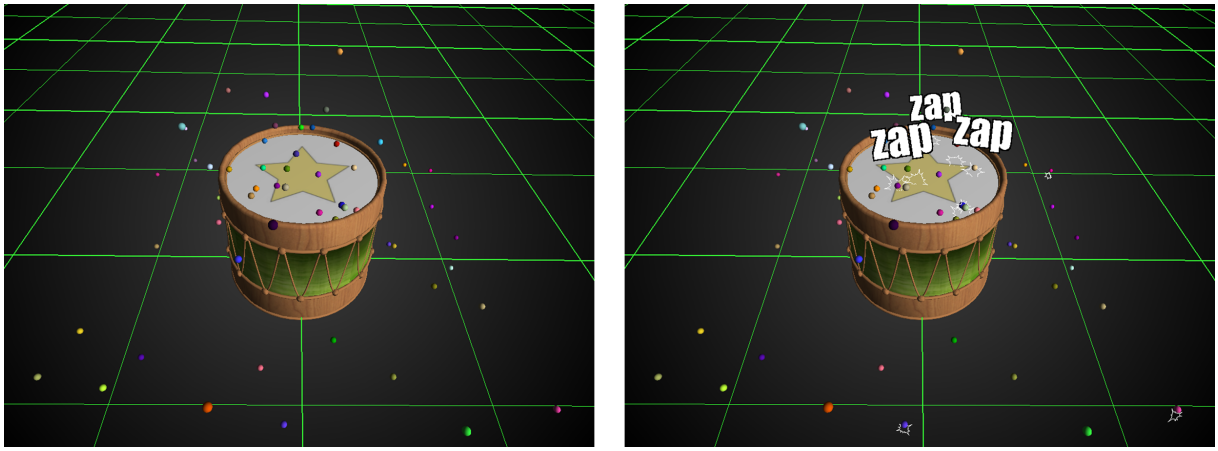


Figure 8: *Comparison of input rigid animation, "drum animation," (left image) with our result (right image). Therefore, the number of times balls collide with drums can be determined, and the force of the collision based on onomatopoeia size can be calculated.*

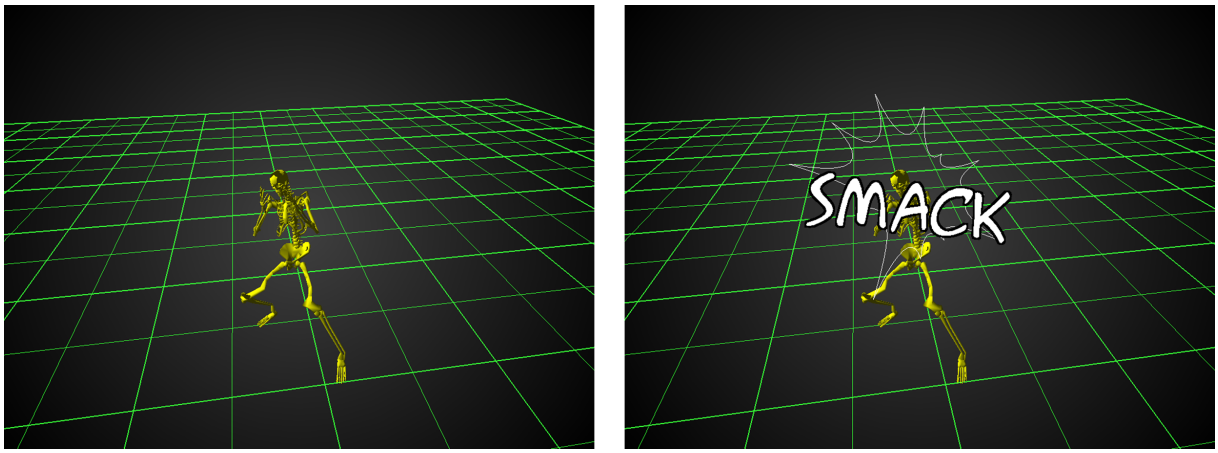


Figure 9: *Comparison of input keyframe animation, "a character get a blow on the head," (left image) with our result (right image). Depicting onomatopoeia enhances the impact of character movement in an animation scene and enables the effective summarization of character movement.*

Acknowledgements

This work was supported by OngaCREST, CREST, JST.

References

BLONK, J., AND LEVIN, G. 2005. Ursonography, *the Art Electronica Festival*.

BLOOMFIELD, L. 1933. Language. *London George Allen & Unwin*, 396-411.

BURR, D.C., AND ROSS, J. 2002. Direct evidence that "Speedlines" inference motion mechanism. *Journal of Neuroscience*, 22,19.

CHADWICK, J.N., ZHENG, C., AND JAMES, D.L. 2012. Precomputed Acceleration noise for Improved Rigid Body Sound. *ACM Transactions on Graphics*, 31, 4, 103.

COLLOMOSSE, J.P., ROWNTREE, D., AND HALL, P.M. 2005. Rendering Cartoon-style Motion Cues in Post production Video. *Graphical Models*, 67, 6, 549-564.

CRYSTAL, D. 1995. Sound Symbolism. *The Cambridge Encyclopedia of the English Language. Cambridge: University Press*, 250-253.

CRYSTAL, D. 2003. A dictionary of Linguistic and phonetics. *fifth Edition, Basil Blackwell*.

CSURKA, G., BRAY, C., DANCE, C., AND FAN, L. 2004. Visual categorization with bags of keypoints. *Proceedings of ECCV Workshop on Statistical Learning in Computer Vision*, 59-74.

DOBASHI, Y., YAMAMOTO, T., AND NISHITA, T. 2003. Real-time Rendering of Aerodynamic Sound Using Sound Textures based on Computational Fluid Dynamics. *ACM Transactions on Graphics*, 22, 3, 732-740.

FORD, S., FORLIZZI, J., AND ISHIZAKI, S. 1997. Kinetic Typography: Issues in time-based presentation of text. *CHI'97 Conference Extended Abstracts*, 269-270.

HONG, R., WANG, M., XU, M., YAN, S., AND SHUA, T. 2010. Dynamic Captioning: Video Accessibility Enhancement for Hearing Impairment. *Proceedings of the international conference on Multimedia*, 421-430.

ISHIHARA, K., TSUBOTA, Y., AND OKUNO, H.G. 2003. Automatic

Transformation of Environmental Sounds into Sound-Imitation words Based on Japanese Syllable Structure. *Proceeding of the Eighth European Conference on Speech Communication and Technology (EuroSpeech 2003)*, 3185-3188.

JACOBSON, R., FANT, G.M., AND HALLE, M. 1965. Preliminaries to Speech Analysis: The Distinctive Feature and Their Correlates. *Cambridge Mass: M.I.T.*

JACOBSON, R., AND WAUGH, L.R. 1979. The Sound Shape of Language. *Brighton The Harvester Press*.

JESPERSEN, O. 1922. Language Its Nature, Development and Origin. *London George Allen & Unwin*.

KHATENA, J. 1969. Onomatopoeia and Images: Preliminary validity study of test of Originality. *Perceptual and Motor Skills*, 28, 1, 335-338.

KOHLER, W. 1929/1947. Gestalt psychology: An Introduction to New Concepts in Modern Psychology. *NewYork: Liveright.*

KOMATSU, T., AND AKIYAMA, H. 2010. Expression System of Onomatopoeia for Assisting Users' Intuitive Expressions. *Transaction on IEICE*, J92A, 11, 752-763 (In Japanese).

KOMATSU, T. 2012. Quantifying Japanese Onomatopoeias: Toward Augmenting Creative Activities with Onomatopoeias. *Proceeding of the 3rd Augmented Human International Conference*, 15.

LEE, D.D., AND SEUNG, H.S. 1999. Learning the parts of objects by non-negative Matrix Factorization. *Nature*, 401, 788-791.

LEE, J.C., FORLIZZI, J., AND HUDSON, S.E. 2002. The Kinetic Typography Engine: An Extensible System for Animating Expressive Text, *Proceedings of the 15th annual ACM symposium on User interface software and technology (UIST'02)*, 81-90.

LEWIS, J.E., AND WEYERS, A. 1999. ActiveText: a method for creating dynamic and interactive texts. *Proceedings of the 12th annual ACM symposium on User interface software and technology (UIST'99)*, 131-140.

LI, D., SETHI, I.K., DIMITROVA, N., AND MCGEE, T. 2001. Classification of general audio data for content based retrieval. *Pattern Recognition Letters*, 22, 5, 533-544.

LYONS, J. 1977. Semantics. *Cambridge: Cambridge University Press*, 1.

MARZAL, A., AND VIDAL, E. 1993. Computation of Normalized Edit Distance and Applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15, 9, 926-932.

MASUCH, M., SCHLECHTWEG, S., AND SCHULZ, R. 1999. Speedline: depicting motion in motionless picture. *Proceedings of SIGGRAPH'99: ACM SIGGRAPH'99 Conference abstracts and applications*, 277.

NARA, Y., KUNITOMI, G., KOIDE, Y., FUJIMURA. W., AND SHIRAI. A. 2013. Manga Generator: Immersive Posing Role Playing Game in Manga World. *Proceedings of Virtual Reality International Conference (VRIC2013)*, 27.

NIENHAUS, M., AND DOLLNER, J. 2005. Depicting Dynamics using principles of visual art and narrations. *IEEE Computer Graphics and Application*, 25, 3, 40-51.

OHALA, J.J. 1994. The Frequency Codes underlies the sound symbolic use of voice pitch. *In Sound Symbolism, Cambridge University Press*, 325-347.

SADOWSKI, P. 2000. The iconicity of English gl-words. *In Olga Fischer and Max Nanny (eds.) The Motivated Sign*, Amsterdam: John Benhamins.

SAPIR, E. 1929. A study in phonetic symbolism. *Journal of Experimental Psychology*, 12, 3, 225-239.

SCHMID, J., SUMNER, R.W., BOWLES, H., AND GROSS, M. 2010. Programmable Motion Effects. *ACM Transactions on Graphics*, 29, 4, 57.

SHOI, M.G., NOH, S.T., KOMURA, T., AND IGARASHI, T. 2013. Dynamic Comics for Hierarchical Abstraction of 3D Animation Data. *The 21st Pacific Conference on Computer Graphics and Applications (Pacific Graphics 2013)*, 32, 7.

TOMOTO, Y., NAKAMURA, T., KANOH, M., AND KOMATSU, T,. 2010. Visualization of Similarity Relationship by Onomatopoeia Thesaurus Map. *IEEE International Conference on Fuzzy Systems(FUZZ)*, 1-6.

UEDA, Y., SHIMIZU, Y., AND SAKAMOTO, M. 2012. System Construction Supporting Communication with Foreign Doctor using Onomatopoeia Expressing Pains. *Joint 6th International Conference on Soft Computing and Intelligent Systems (SCIS) and 13th International Symposium and Advanced Intelligent System(ISIS)*, 508-512.

ULLMAN, S. 1962. Semantics: An Introduction to the Science of Meaning. *Oxford: Basil Blackwell*.

UMEDA, D., MORITA, T., AND TAKAHASHI, T. 2012. Real-time Manga-Like Depiction Based on Interpretation of Bodily Movements by Using Kinect. *Proceedings of ACM SIGGRAPH Asia 2012 Technical Briefs*, 28, 1-4.