

TextAlive:

インタラクティブでプログラマブルな
Kinetic Typography制作環境

加藤 淳 中野 倫靖 後藤 真孝
産業技術総合研究所

Kinetic Typographyとは

この世界のメロディー

Kinetic Typographyとは

- 静止したテキストよりも、文字情報を印象的に見せられる映像表現の一手法
- **コミュニケーションツール**への応用研究が多い

[Lee, 2006] [Forlizzi, 2003] [水口, 2005] [Strapparava, 2006]など

本研究では、とくに

文字が音楽の歌詞と同期して動くものを扱う



時間軸の制約が加わり、動画制作が難しくなる

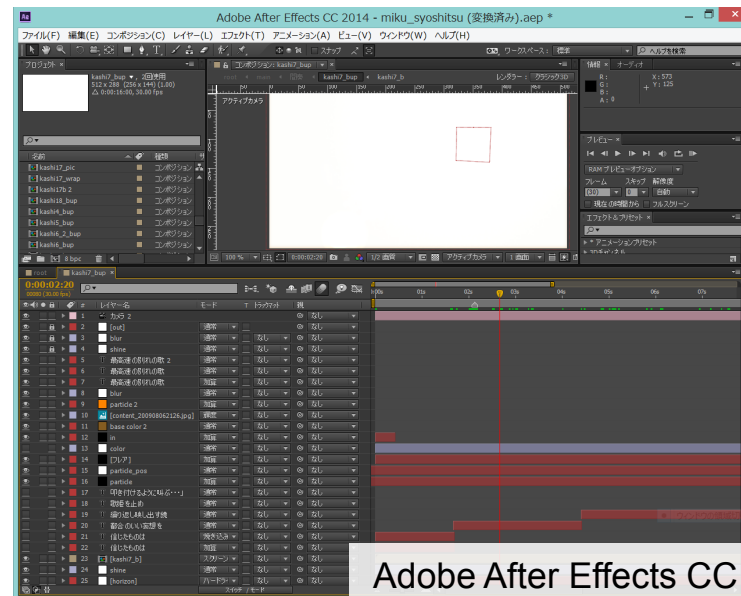
汎用動画制作ツールでの動画制作

☹️ **Typography**の要素
を**文字毎**に指定

例) 字形、サイズ、色…

☹️ **動き**を**ゼロから**設計

例) 出現タイミング、軌跡…



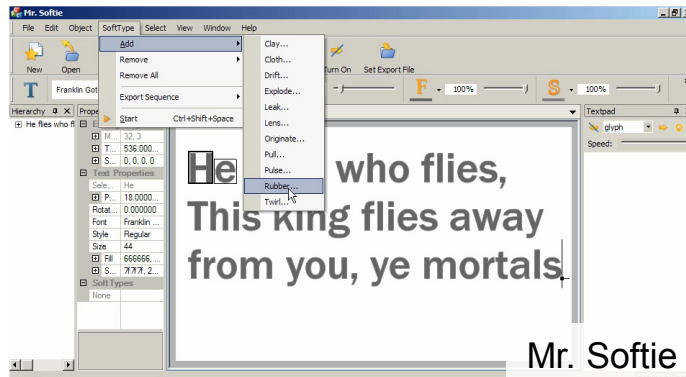
膨大な手作業

専用動画制作ツールでの動画制作

😊動きをテンプレートから
選んでパラメタ調整



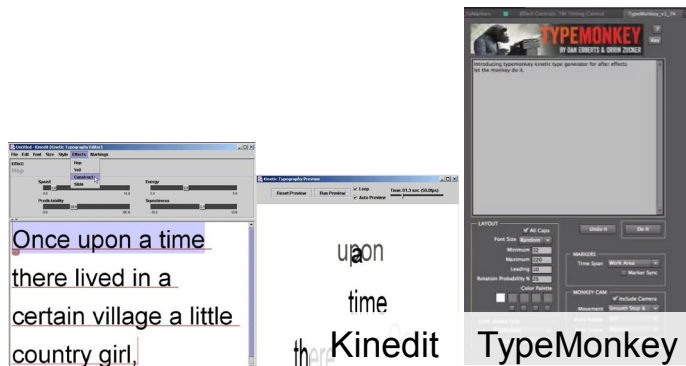
自由度に限界



😞発声タイミングを各文字
字に手動で設定



膨大な手作業



TextAlive

問題

膨大な手作業

自由度に限界



提案手法



動画の自動生成と
インタラクティブな編集

テンプレートの
Live Programming

インタラクティブで

プログラマブルな

Kinetic Typography制作環境

TextAlive:

インタラクティブでプログラマブルなKinetic Typography制作環境

動画の自動生成と インタラクティブな編集

Project Song Options

Selected time region
Start: - End: - Auto sync

Selected text units
10 units

Font
Style: - Size: 0.0

Animation
Char Dropping (No animation specified)

Remove Edit

(No animation specified)

Play Stop 100 0 25 50 75 100

Phrases
この世界のメロディー わたしの歌声 届いているかな 響いているかな この世界のメロディー わたしの歌声 届いている

Words
この 世界 の メロディー わたし の 歌声 届いて いる かな 響いて いる かな この 世界 の メロディー わたし の 歌声 届いて

Characters
この 世界 の メロ - たし の 歌 声 届 いて いる かな 響 いて いる かな この 世界 の メロ - たし の 歌 声 届 いて

Graphics

Chorus

・発声タイミングの自動推定
・インタラクティブなタイミング訂正

←00:02:78 00:31:70→

発声タイミングの自動推定

TextAlive ver.a 本プロジェクトについて 使い方

【メグuppoid】 Packaged 【ばかりす + ぼかうお】

アーティスト: VocaListener (AIST)
 楽曲リンク: <http://staff.aist.go.jp/t.nakano/music/VocaWatcher.Packaged.Megpoid.mp3>
 歌詞リンク: <http://www5.atwiki.jp/hmiku/pages/16.html>

歌詞

+

音楽

→

まだ作例はありません。

Kinetic Typography動画を作ってみる

© 2014 AIST TextAlive Project

TextAlive ver.a 本プロジェクトについて 使い方

Project Song Options

この世界のメロデー

Selected time region

Start: -
End: -

Selected text units

No unit selected

Font

Style: (No particular font selected)
Size: []

再生可能な動画

Play Stop [Speaker icon] 0 25 50 75 100 iphy

Phrases: この世界... わたし... 聞いて... 響いて... この世界... わたし... 届け

Words: こ × な 歌 声 届 い 響 い こ × な 歌 声 届

Characters: 声 声 声

Graphics: []

Chorus: []

← 00:05:44 00:32:23 →

発声タイミングを手動で設定する手間を削減

インタラクティブなタイミング訂正

この世界のメロディー
わたしの歌声

Phrases

この世界の... わたしの歌声

W

タイミング訂正

歌声

Characters

世界 歌声

Graphics

Chorus

←01:40:57

この世界のメロディー

再生中の動画に反映

タイミング自動推定の誤りも容易に訂正可能

Project Song Options

Selected time region
Start: - End: - Auto sync

Selected text units
10 units

Font
Style: - Size: 0.0

Animation
Char Dropping (No animation specified)

Phrases	この世界のメロディー	わたしの歌声	届いているかな	響いているかな	この世界のメロディー	わたしの歌声	届いている
Words	この 世界 の メロディー	わたし の 歌声	届いて いる かな	響いて いる かな	この 世界 の メロディー	わたし の 歌声	届いて
Characters	この 世界 の メロ	ー わたし の 歌 声	届 いて いる かな	響 いて いる かな	この 世界 の メロ	ー わたし の 歌 声	届 いて

・文字・単語・フレーズごとの編集

←00:02:78 00:31:70→

文字・単語・フレーズごとの編集

Sliding

この世界のメロディー

Sliding + Hopping

この世界のメロディー

Sliding + Hopping + Karaoke

この世界のメロディー

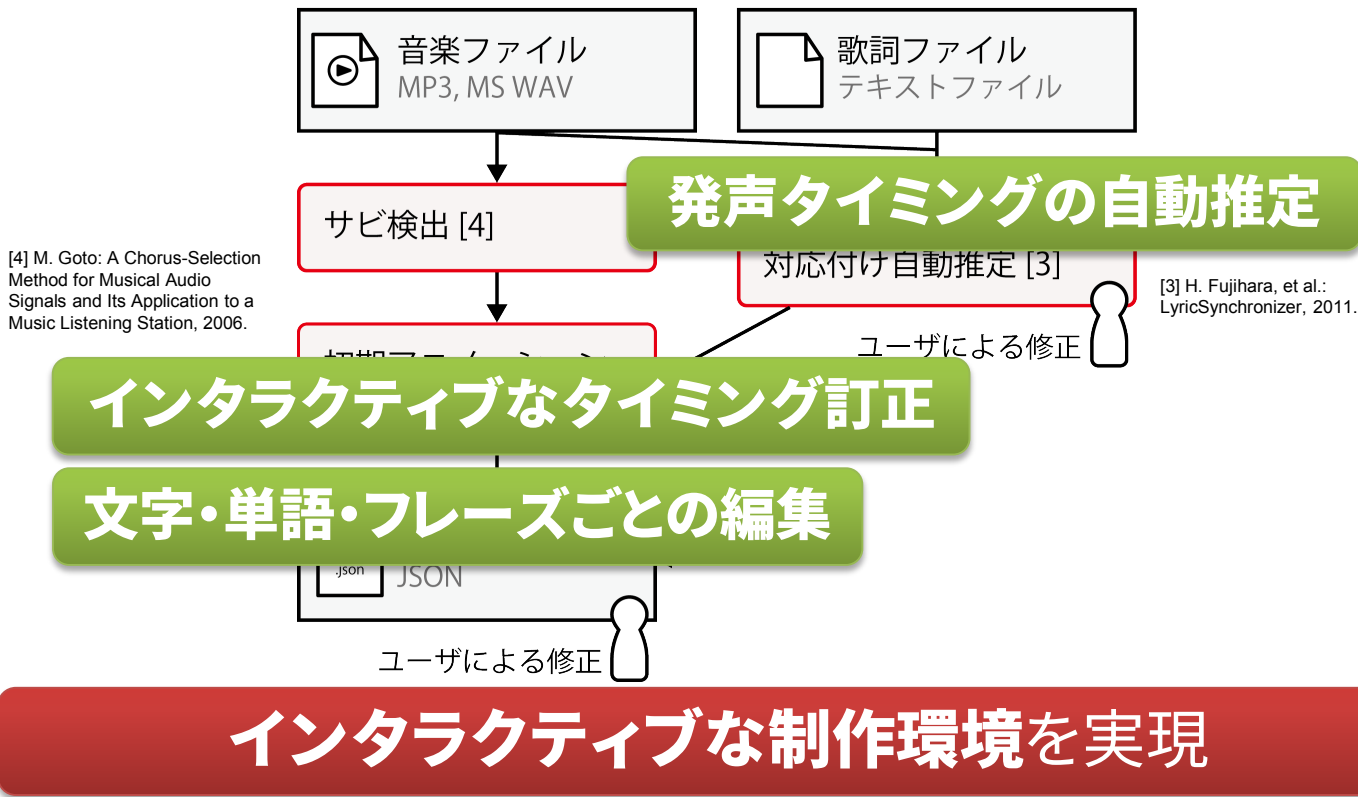
歌詞の構造を利用して見た目や複雑な動きを設計

文字・単語・フレーズごとの編集



歌詞の構造を利用して見た目や複雑な動きを設計

TextAliveの処理の流れ



TextAlive:

インタラクティブでプログラマブルなKinetic Typography制作環境

テンプレートの LIVE PROGRAMMING

Project Song Options

Selected time region
Start: - End: - Auto sync

Selected text units
10 units

Font
Style: - Size: 0.0

Animation
Char Dropping (No animation specified)

Remove Edit

(No animation specified)

Play Stop 100 0 25 50 75 100

Phrases
この世界のメロディー わたしの歌声 届いているかな 響いているかな この世界のメロディー わたしの歌声 届いている

Words
この 世界 の メロディー わたし の 歌声 届いて いる かな 響いて いる かな この 世界 の メロディー わたし の 歌声 届いて

Characters
この 世界 の メロ - たし の 歌 声 届 いて いる かな 響 いて いる かな この 世界 の メロ - たし の 歌 声 届 いて

Graphics

Chorus

・テンプレートのLive Programming

←00:02:78 00:31:70→

Live Programming

The screenshot displays the TextAlive ver. alpha interface. At the top left, it says "TextAlive ver. alpha". To the right, there are links for "本プロジェクトについて" and "使い方". Below this is a "Project Song Options" section. The main area is a "Code Editor: SlidingAnimation2" showing JavaScript code for a sliding animation. The code includes a function definition for SlidingAnimation2, a getName method, an isAssignable method, and a rotationSpeed property. A double-headed arrow connects two purple boxes: "動画制作" (Video Production) on the left and "テンプレート開発" (Template Development) on the right. A diagonal red text "界のメロデー" (Melody of the World) is overlaid on the arrow.

```
Code Editor: SlidingAnimation2
1 function SlidingAnimation2() {
2
3 this.getName = function() {
4
5
6
7 this.isAssignable = function(unit) {
8   return unit.getType() == "Phrase";
9 };
10
11 /**
12  * Rotation speed [degree/second]
13  * @ui Slider(0, 360)
14  */
15 this.rotationSpeed = 20;
16
```

コンテンツ制作とプログラミングをスムーズに往復

Live Programmingの実現方法

- テンプレートは**クラス**として実装
- 状態情報は**publicフィールド**にあると仮定

「Save」ボタンを押した時の処理(概要):

編集中のテンプレートを利用している
アニメーションのシリアライズ(JSON化)

テンプレートインスタンスおよびクラス情報の破棄

新クラスのコンパイルとクラスローダーによる読み込み

アニメーションをJSON表現からデシリアライズ
(状態情報の復元はベストエフォート)



The screenshot shows a code editor window titled "Code Editor: ShakeAnimation". The code is in Java and includes package declarations, imports for java.awt.*, java.awt.geom.*, and custom packages for data and image. It defines a public class ShakeAnimation that extends AbstractAnimation. The code includes comments for a slider and a vibration field.

```
package jp.go.aist.textalive.animation;

import java.awt.*;
import java.awt.geom.*;
import jp.go.aist.textalive.data.*;
import jp.go.aist.textalive.image.*;

public class ShakeAnimation extends AbstractAnimation {

    /**
     * @ui Slider(0..100)
     */
    private int vibration = 10;

    public int vibration = 10;

    Name() {
    }
}
```

一行コメントによる制作環境GUIの拡張

@ui Slider(最小,最大値)

offset: 2339



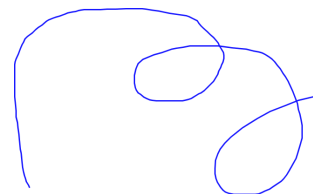
@ui Radio(項目,値,...)

horizontalAlignment:

left center right

@ui Track()

Draw track



プログラマは…

各テンプレートの**デバッグ**を容易に行える

エンドユーザは…

各テンプレートの**パラメタ調整**を容易に行える

Project Song Options

The screenshot displays the TextAlive software interface. The main workspace shows a large, stylized Japanese title 'アモ' (Amo) in black, with a subtitle '世界のメロディ' (World's Melody) in white. The background is a light blue gradient with bokeh effects. A control panel at the bottom includes a play button, a stop button, a volume slider, and a timeline. The timeline shows a selected time region from 00:02:78 to 00:31:70. A text table is visible below the timeline, and a right-hand panel shows settings for the selected text units, including font and animation options.

Phrases	この世界のメロディー	わたしの歌声	届いているかな	響いているかな	この世界のメロディー	わたしの歌声	届いている
Words	この	世界	の	メロディー	わたしの	歌声	届いて
Characters	こ	の	世	界	の	メ	ロ
Graphics	[Blue bar]						
Chorus	[Orange bar]						

•一行コメントによるGUI拡張

TextAliveの処理の流れ



[4] M. Goto: A Chorus-Selection Method for Musical Audio Signals and Its Application to a Music Listening Station, 2006.

[3] H. Fujihara, et al.: LyricSynchronizer, 2011.

テンプレートのLive Programming

一行コメントによる制作環境GUI拡張

プログラマブルな制作環境を実現

TextAlive:

インタラクティブでプログラマブルなKinetic Typography制作環境

おわりに

TextAlive

Project Song Options

Selected time region

Start: . End: . Auto sync

Selected text units

10 units

Code Editor: ShakeAnimation

```
1 package jp.go.aist.textalive.animation;  
2  
3 import java.awt.*;  
4 import java.awt.geom.*;  
5 import jp.go.aist.textalive.data.*;  
6 import jp.go.aist.textalive.image.*;  
7  
8 public class ShakeAnimation extends AbstractAnimation {
```

インタラクティブで

プログラマブルな

Kinetic Typography制作環境

Play

Phrases

Words

Characters

Graphics

Chorus

00:03:21 00:50:19+

Shake

speed: 33

vibration: 16

Save Close

TextAlive

Project Song Options

Selected time region

Start: ...

End: ...

Selected text units

Code Editor: ShakeAnimation

```
1 package jp.go.aist.textalive.animation;
2
3 import java.awt.*;
4 import java.awt.geom.*;
5 import jp.go.aist.textalive.data.*;
6 import jp.go.aist.textalive.image.*;
7
8 public class ShakeAnimation extends AbstractAnimation {
```

甲のメロディー

インタラクティブで プログラマブルな

<http://textalive.jp>

Kinetic Typography制作環境

Shake

speed: 33

vibration: 16

本日公開!

未来ビジョン



統合開発環境(IDE)は超☆便利!

```
$ vi MyCode.java
```

Vim? Emacs? ……CUI??

***.viml? *.el? Atom?**

Programmability = いじりやすさ



TextAlive

The screenshot displays the TextAlive software interface. On the left, a 'Project Song Options' window shows a blue background with the text '甲のメロディー'. Below it, a timeline with 'Phrases', 'Words', 'Characters', and 'Chorus' tracks is visible. A green callout bubble contains the text 'インタラクティブで'. On the right, a 'Code Editor: ShakeAnimation' window shows Java code for an animation class. A purple callout bubble contains the text 'プログラマブルな'. A large black callout bubble at the bottom center contains the text 'Kinetic Typography制作環境'.

インタラクティブで

プログラマブルな

Kinetic Typography制作環境

<http://textalive.jp>