

TextAlive:

Integrated Design Environment for Kinetic Typography

Jun Kato, Tomoyasu Nakano, Masataka Goto
AIST, Japan

Kinetic Typography

この世界のメロディー

Kinetic Typography

- A style of graphical expression to **move and reshape text** in videos
- Prior work focuses on **graphical** aspects, most augmenting text-based communication

[Lee, 2006] [Forlizzi, 2003] [Mizuguchi, 2005] [Strapparava, 2006]

This work focuses on

Animating **lyric text** in synchrony with **music**



Synchronization is difficult & takes long time

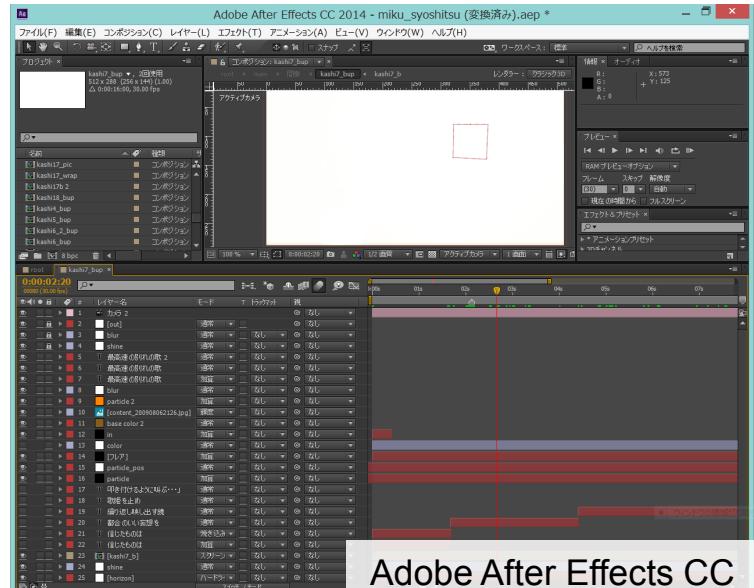
Authoring with **general tools**

⌚ Set **typographic parameters per character**

e.g. Font, Size, Color, ...

⌚ Specify **kinetic behavior from scratch**

e.g. Timing, Motion path, ...



Enormous amount of manual effort

Authoring with **specialized tools**

⌚ Set **typographic parameters per character**

e.g. Font, Size, Color, ...

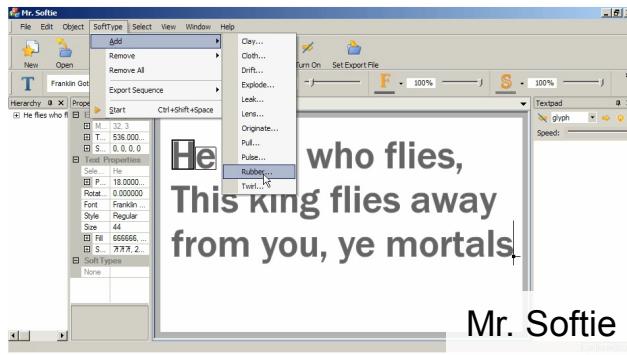
⌚ Specify **kinetic behavior from scratch**

e.g. Timing, Motion path, ...

⌚ Choose motion from **templates** and tweak parameters

Authoring with **specialized tools**

☺Choose motion from **templates**



☺Manual labor remains ☺Lack of flexibility
to specify lyric timings

should be more **effortless** and **flexible**

TextAlive

Problem

Manual effort

Lack of flexibility

Proposed Method

Automatic Video
Composition

Live Programming
of Templates

Interactive
Editing

For designers

For programmers

Integrated Design Environment

TextAlive:

Integrated Design Environment for Kinetic Typography

**AUTOMATIC VIDEO COMPOSITION
+ INTERACTIVE EDITING**

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

Play Stop 100 0 25 50 75 100

Kinetic Typography for Everyone

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
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
この の 世界 の メロディー	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	たしの 歌 声	届いて

Words

Characters

Graphics

Chorus

←00:02:78 00:31:70→

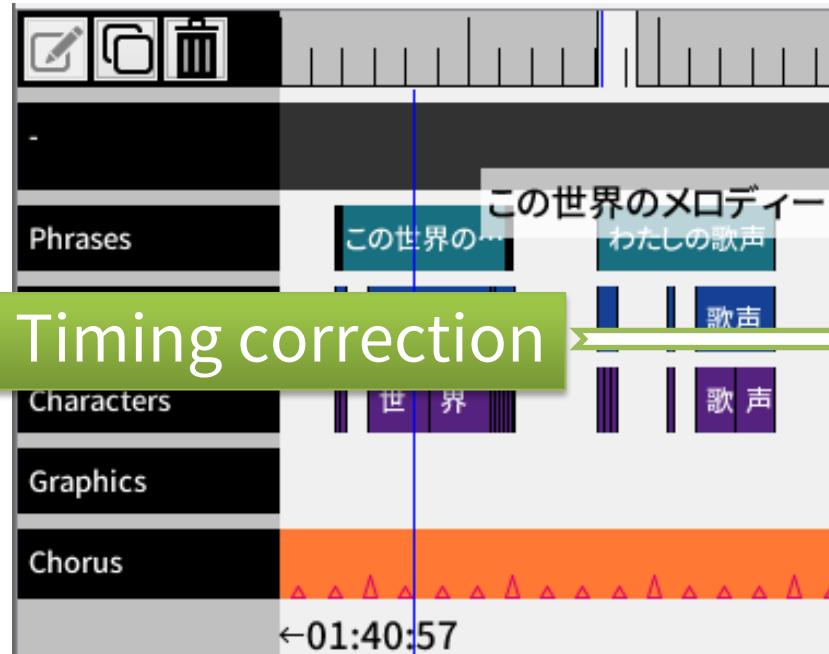
- Automatic timing estimation
- Interactive timing correction

Automatic timing estimation



drastically reduces the manual effort

Interactive timing correction



Estimation errors can be easily corrected

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)

(No animation specified)

Play Stop 100 0 25 50 75 100 Remove Edit

Kinetic Typography for Everyone

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
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
この の 世界 の メロディー	たしの 歌 声	届いて いる かな	響いて いる かな	この の 世界 の メロディー	たしの 歌 声	届いて

Words

Characters

Graphics

Chorus

•Structure-aware editing

←00:02:78 00:31:70→

Structure-aware editing

Sliding

この世界のメロディー

Sliding + Hopping

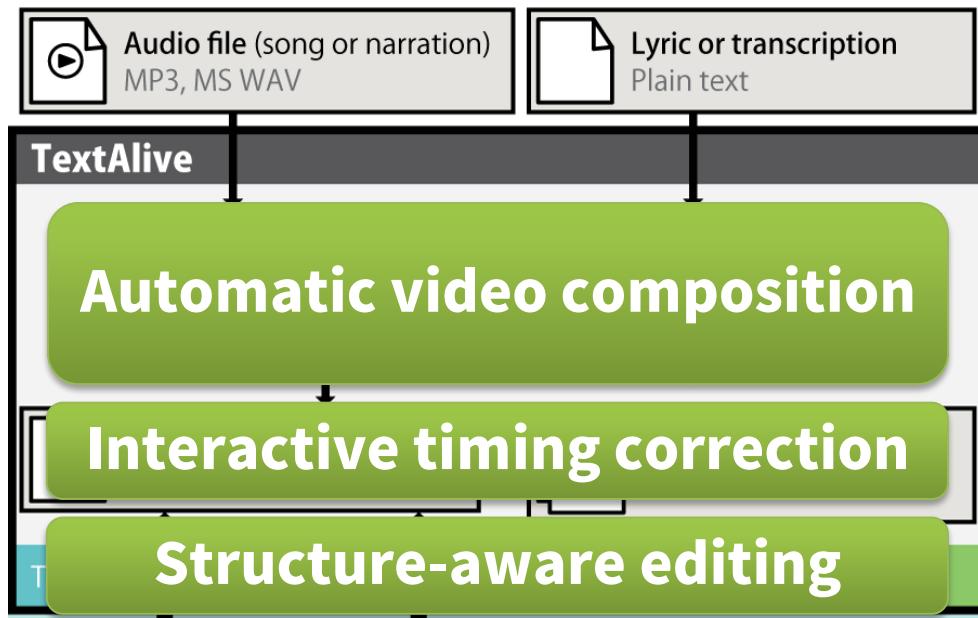
この世界のメロディー

Sliding + Hopping + Karaoke

この世界のメロディー

Multiple templates can be assigned to
phrases/words/characters

TextAlive workflow (for designers)



→ : Data flow or dependency

↔ : User interaction

TextAlive workflow (for designers)



Am I in the wrong session? All of these **fancy GUIs**,
How are they related to the “**Programming Environments**”???



Interactive editing of Kinetic Typography videos

→ : Data flow or dependency

↔ : User interaction

TextAlive:

Integrated Design Environment for Kinetic Typography

LIVE PROGRAMMING OF ANIMATION TEMPLATES

Video = $f(time)$ without side effects

1. for (all templates)

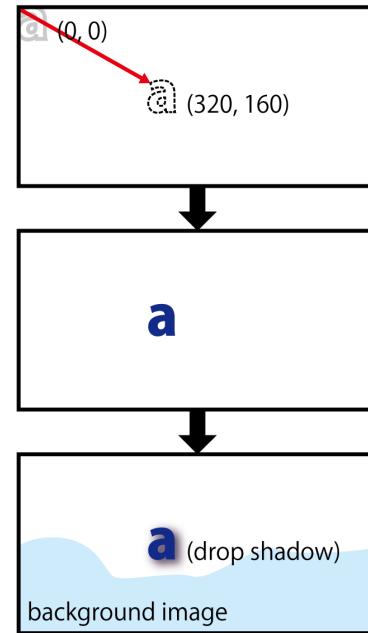
- manipulate rendering parameters of assigned text units

```
35 // Make this char hop!
36 float progress = Math.min((float)(time - c.startTime)/hopTime, 1f);
37 c.rendering.tx.translate(
38     0, (float)-hopHeight * Math.sin(progress*Math.PI));
```

2. for (phrases) transform the coordinate system

- for (words) transform the coordinate system
 - for (characters) transform the coordinate systems;
 - render the character.

3. for (graphic units) render graphics



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

Play Stop 100 0 25 50 75 100

Kinetic Typography for Everyone

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
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
この の 世 の メロ ディー	たし の 歌 声	届 い て い る か な	響 い て い る か な	この 世 の メロ ディー	たし の 歌 声	届 い て

Words

Characters

Graphics

Chorus

←00:02:78 00:31:70→

• Live programming of animation templates

Live Programming



Dynamic generation of GUI widgets

```
21 //  
22 * @ui Slider(0, 1000)  
23 */  
24 public int hopTime = 200;
```



Comments to **public fields** generates GUI widgets bound to the state information

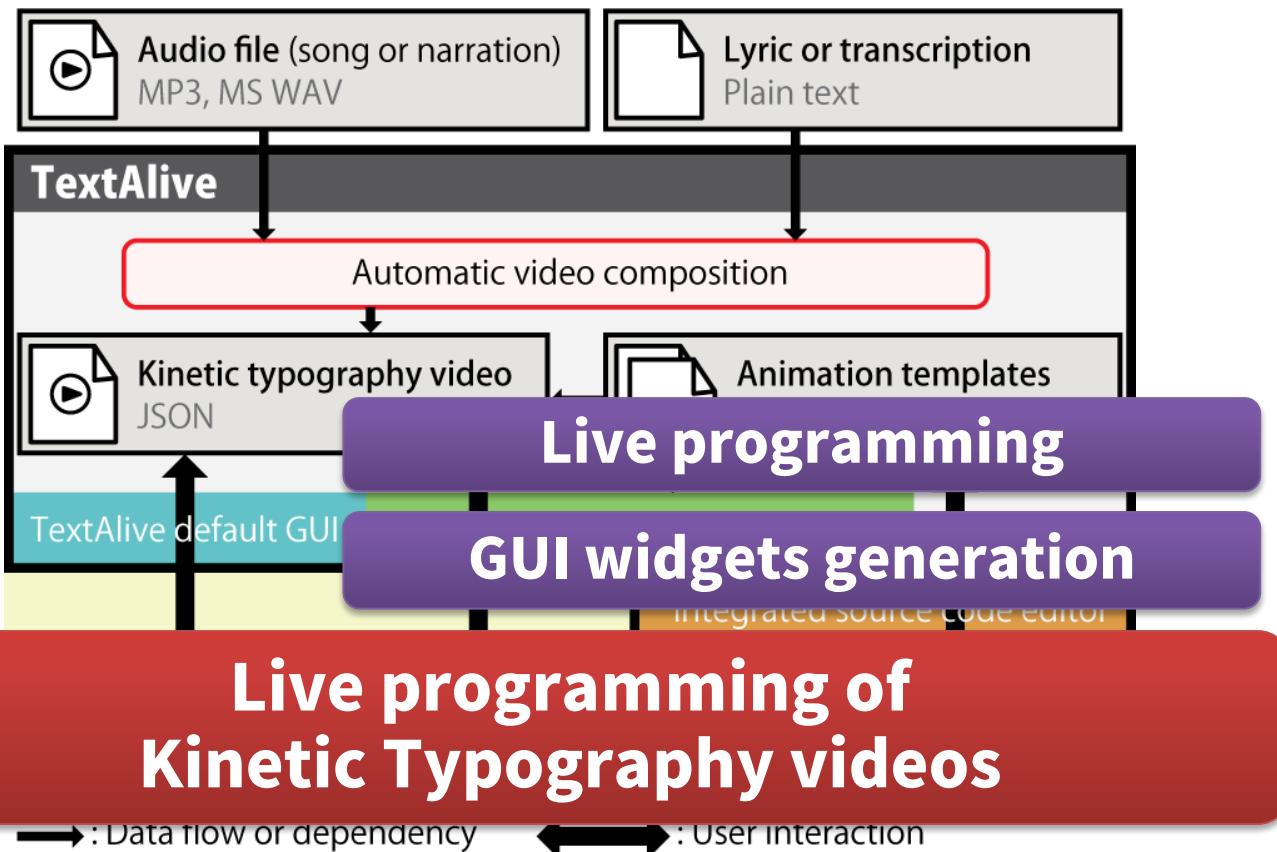
for programmers:

enables easy template **debugging**

for designers:

enables interactive **parameter tuning**

TextAlive workflow (for programmers)



TextAlive:

Integrated Design Environment for Kinetic Typography

PRELIMINARY USER STUDY

Experimental setup



Designers and casual users

Programmers

Amateur singer, illustrator, disk jockey, interaction designer



3 have lots of experience, **2** have a little bit of experience,
2 have no experience of using video authoring tools.

Each participant was asked to create a video

Results



Lessons learned: good side ☺

- Automatic video composition is nice since “*we no longer need to create but just edit the video.*”
- Animation templates suggest potential applications of real-time video creation such as “*text jockey.*”
- No need to stop the video = fluid experience
- “Timeline” at the heart of audio-video synchronization

especially for programmers

Integrated interfaces worked as expected

(this should be of interest if you make something similar)

Lessons learned: future work ☺

- Having a variety of animation templates is good
- “Stage” should allow more direct manipulation
- (Programming needs training)

Collaboration between designers and programmers seems to be the key

(and should be an interesting follow-up work)

TextAlive:

Integrated Design Environment for Kinetic Typography

CONCLUSION

TextAlive

The screenshot shows the TextAlive application window. On the left, a preview window displays a blue background with Japanese text "この世界のメロディー" (Kono Sekai no Melody) in white, with red lines indicating text paths. To the right are several panels: "Selected time region" (Start: -, End: -, Auto sync), "Selected text units" (10 units), "Font" (Style: [dropdown], Size: 0.0), "Animation" (Char Dropping (No animation specified)), and "Code Editor: ShakeAnimation". The code editor contains Java-like pseudocode for a ShakeAnimation class.

Automatic Video Composition

For designers

Interactive Editing

Live Programming of Templates

For programmers

Integrated Design Environment

TextAlive

(completely rewritten in JavaScript!)

The screenshot shows the TextAlive application interface. On the left, a video player displays a scene from a movie with Japanese text overlaid. A green callout bubble points to this area with the text "Automatic Video Composition". Below it, another green callout bubble points to the "Word" tab in the bottom navigation bar with the text "For designers". In the center, there's a control panel with "Selected time region" and "Selected text units" dropdowns, and a "Char Dropping" section. A red callout bubble points to this area with the text "Interactive Editing". To the right, a code editor window titled "Code Editor: ShakeAnimation" shows Java-like pseudocode for an animation. A purple callout bubble points to this area with the text "Live Programming of Templates". Below the purple callout, a purple speech bubble points to the "Code" tab in the bottom navigation bar with the text "For programmers". At the bottom, a large black button contains the text "Integrated Design Environment".

Automatic Video Composition

For designers

Interactive Editing

Live Programming of Templates

For programmers

Integrated Design Environment