

Development of Primitive Motion Library for Kinetic Typography Rendering Engine

YoonAh Cho, JooHyun Park and Soon-Bum Lim*

*Dept. Multimedia Science, Sookmyung Women's University, Korea
fanwith@naver.com, park.Joohyun5@gmail.com, sblim@sookmyung.ac.kr*

Abstract

Kinetic Typography Rendering Engine contains Primitive Motion Library to convey the emotions. It wraps essential elements such as time, space and speed to express movement. Scrutinized primary movement is constituted as Primitive Motion Library and it is a way to kinetic typography properly. The main purpose of Primitive Motion Library substitutes the existing method to manufacture kinetic typography; such as using software and doing expert coding. To achieve this purpose, the thesis is separating the movement, classifying it systematically and developing Primitive Motion API which renders Kinetic typography.

Keywords: *Kinetic Typography, Primitive Motion, Motion Library, Primitive Motion Library, Motion API, Motion Component, Systematic Primitive Motion API, Rendering Engine*

1. Introduction

Kinetic typography is a medium to go forward a moving text. It indicates text movement by adding up the variety of effects of the moving text such as time, space, speed and weight to show both temporal and spatial moving. The first serious analysis about kinetic typography emerged in 1995 from temporal typography. In addition, a technical research is progressed in MIT and CMU; implementation of the engine for kinetic typography [1] and production of time-based technology [2] are distinctive researches. Kinetic typography can be utilized as a tool of communication in company with the rapid development of visual culture and digital technology. It infers that kinetic typography can become used to numerous application systems and transition the texts. In spite of its encouragement and expressiveness, kinetic typography encompasses several major drawbacks. Most studies in the field of kinetic typography have examined in aesthetic signification and creative presentation. Furthermore we should use professional animation software such as Flash [9], After Effect [10] and other inconvenient programming stage to generate kinetic typography. These are time-consuming tasks or requiring professional skill. Additionally it cannot illustrate kinetic typography in the environment of social media. This paper will constitute Primitive Motion Library using Primitive Motion API to replace the inconvenience. There are two primary objectives of this study: 1) to evaluate the representative motion systematically 2) to supersede existing method with Primitive Motion Library.

* Corresponding Author

2. Related Works

2.1 Preceding Framework: Engine for Kinetic Typography

Studies on technical method of kinetic typography have been conducted since the late 1990s. Kinetic typography engine [3] is a foundational framework for generating kinetic typography; it provided basic maneuvers such as moving, rotating and twisting of the texts. Recently, KTE2 [4] was published with its additional features. However, the engine was inappropriate for authoring tools in Korean kinetic typography and unequipped with the interface for it.

2.2 Preceding Research: Mobile Instant Messenger Using Kinetic Typography

Mobile Instant Messenger Using Kinetic Typography [8] was designed to create kinetic typography quickly and easily in any users' skill and proficiency. In the process, the Typography analyzed, organized and divided the motion into four groups of motion libraries. Primitive Motion API reanalyzed the motion with the systematic classification based on the precedent research.

3. Model of Primitive Motion Library

3.1 Purpose of Primitive Motion Library

There are lots of differences between animation authoring tools and kinetic typography authoring tools. If people who want to express the movement freely and have production time sufficiently configure kinetic typography, existing animation authoring tools can be suitable method for them because several authoring tools for the animation are widely used to illustrate the Media Art. However, it is becoming increasingly difficult to ignore the environment of Social Media. In social media environment, the way to express kinetic typography should be fast and easy. It also has restrictions of screen size and motion moved. Primitive Motion API will be able to utilize the kinetic typography in the social media environment effectively.

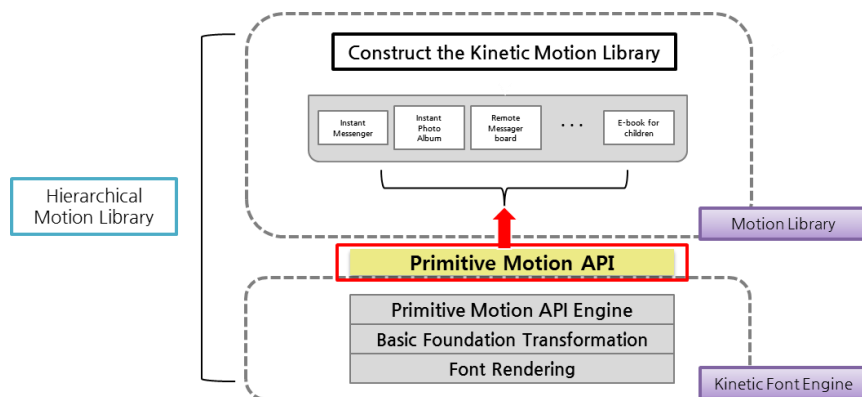


Figure 1. Hierarchical Motion Library applied Primitive Motion API

Primitive motion API is defined as the fundamental level for controlling the geometric transformation of the kinetic motion. It provides a function of motion that is the basis of the motion library in accordance with the domain. Besides, kinetic font engine includes font rendering module to show vector font applied to the kinetic motion effect on the screen that can easily be configured to provide the kinetic motion library according to the domain. Figure

1 shows kinetic font engine via hierarchical motion library to afford Primitive Motion API in standard web environment. Using primitive motion API should construct the kinetic motion library and refer to diverse applications.

3.2 Classification of the Motion

The construction of kinetic typography is distinguished from the production with existing PC-based animation manufacture tools in its implication and environment. Existing PC-based animation programs are often utilized for art or media art, requiring professional and sophisticated skills. Consequently it is relatively unchained by restrictions as screen size and time. However, it is bound in mobile environment; for example, the range of motion and the method to select and to set up the menu are restricted in mobile environment. Thus, problems on expressing motions will occur with existing PC-based animation manufacture tools in social-media background. In consequence, the paper proposes primitive motion API to be utilized efficiently in the social-media environment.

3.2.1 Combination of the Motion

In this paper, it evaluates PC-based kinetic typography manufacturing tools of Adobe's Flash, AfterEffect, and animation effects on Microsoft's PowerPoint and Apple's KeyNote. As a result, the research classifies the components of kinetic typography into two major parts: general factors and motion factors, which are further categorized into 6 general factors and 7 motion factors each. General factors are composed of font, size, color, underline, italics and space; motion factors are comprised of opacity, speed, direction, location, separation, size and repetition. The classification system referenced SMIL Transition to categorize kinetic typography motions.

According to the way for the typography to move, motion of kinetic typography is classified into three basic motions. Each motion is grouped in sub-motions; basic motions are classified into static motion, straight motion, curved motion and free motion and are decided by the attribute values speed, separation, rotation, size of motion, size of letter, and value of opacity which influences the motion. Static motion lacking direction does not have sub-motion; straight motion has sub-motions distinguished by right directions of up, down, right and left and oblique directions of the angle of divergence; curved motion has sub-motions differentiated by shape of motion basically into circular motions; free motion lacking regularity does not have sub-motion.

3.2.2 Formation of the Primitive Motion API







Primitive Motion API is generally classified into static motion, straight motion and curved motion and subdivide into sub-motion APIs like Table 1. Also, each API has its attribute value to express the motion.

Table 1. Design the Primitive Motion API

Primitive Motion API = { <Static Motion>, <Straight Motion>, <Curved Motion> } Sub-motion API of Static Motion = { <Static Rotation>, <Static Shaking>, { <Static Opacity Change>, <Static Scale Change>} Sub-motion API of Straight Motion = { <Straight Movement> } Sub-Motion API of Curved Motion = { <Circle Movement> }

Table 2 arranges the attribute values of each sub-motion API; the arrangement is to enable users to produce kinetic typography simply by selecting and changing the motion attribute values.

Table 2. Example of Primitive Motion API and Parameters

Static Rotation : { object, angle, duration time } 	Static Shaking : { object, size, duration time } 
Static Opacity : { object, opacity value, duration time } 	Static Scale : { object, size, duration time } 
Line : { object, direction, length, duration time } 	Circle : { object, angle, radius, duration time } 

Each departmentalization API varies according to alteration in the properties of its value. For instance, to set the circular movement into the text object named A declares the Circle API to A. Then to adjust other properties such as rotation angle, radius and duration time makes complete kinetic typography to circular movement of A.

4. Implement of Primitive Motion Library

Primitive Motion API is designed by extracting the properties of each group and building templates for all groups. It can display movement which is applied in application. Primitive Motion API is developed in JavaScript Library which is the basis for standard web environment. User should input the library into the web and mobile application, and call the API to element of HTML file made by user. Its aim is providing method to generate kinetic typography for not the experts but also ordinary people. They articulate movement into texts using Primitive Motion API quickly and easily.

4.1 Structure of Primitive Motion API

Primitive Motion API controls geometrical movements including Font Rendering Module and Movement Controlling Module. It is designed by forming the text object, setting up the attribute values of the texts and creating it as a function. Primitive Motion API can be used by calling.

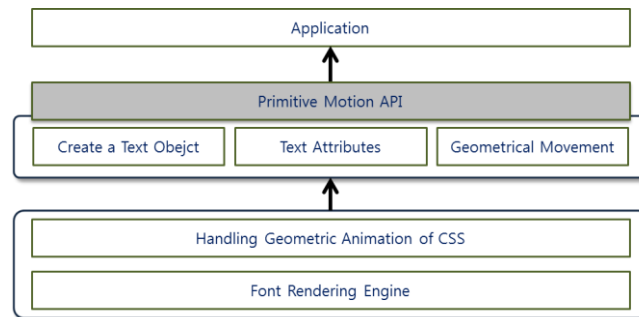


Figure 2. Structure of Primitive Motion Library

4.2 Usage of the Primitive Motion Library

The Interface is formulated to regulate paths and properties of each motion. It is particularly applicable to PC, mobile phone and tablet PC. And it is also practical to application such as web, Mobile messenger, advertising contents and documents. The interface will be used to convey emotions with time synchronized technique among multi objects in real-time. First of all, the interface calls Setting API to establish basic resources of the text as properties of typography (contents, font-size, font-color and opacity) and initial values (starting position of motion, angle, *etc.*). Then it calls other Primitive Motion API such as Static Motion API, Straight Motion API and Curved Motion API. It displays the motion by setting text object and properties of the motion. Also it shows source code to call the API; therefore, the user utilizes code again. To provide Multi-Platform Environment such as PC • mobile phone • Tablet PC, primitive motion API is made in JavaScript based on HTML. Each API is called as embedded or plug-in and attached to elements of HTML which are made by user.

A person, who wants to configure kinetic typography by using the primitive motion API, should add primitive motion library into his HTML file first and generate elements with a unique identifier. If he calls the primitive motion API, the API will transfer text object to movement object. Lastly, he can adjust API's properties.

Figure 3 shows the interface for developers. Developers can modify properties of each primitive motion API and check the motion intuitively. In addition, developers are able to use the preset motion in his application.

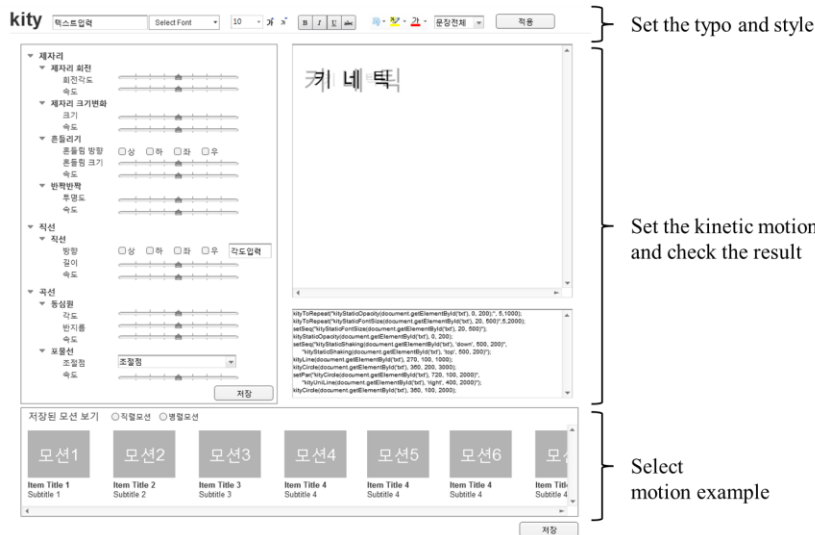


Figure 3. Interface of Primitive Motion Library for Developer

5. Usability Evaluation of Primitive Motion Library

In this chapter we verify usability of primitive motion library by doing the user evaluation. For this step, we exploit the Mobile Postcard Application using primitive motion API and manufacture the application not using primitive motion library but also using SVG language and then compare the application of primitive motion API and the application of SVG language to show effectiveness and efficiency of primitive motion library. Also, the application demonstrates the example of primitive motion library in actual world.

5.1 Development of the Mobile Postcard Application

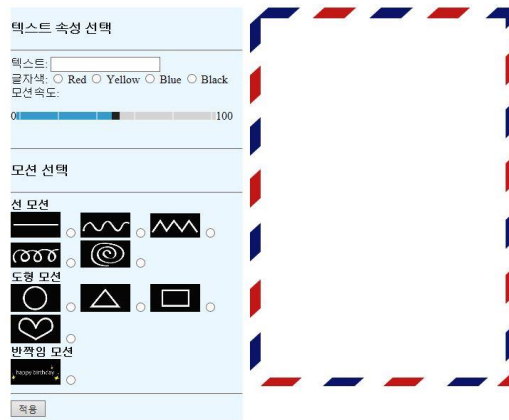


Figure 4. Development the Kinetic Postcard Application



The motion library of the application is comprised of three motion groups such as linear motion group, shaped motion group and sparkling motion group. The movement in linear motion group and shaped motion group shows the text movement by following the path and sparkling motions changes the value of opacity. The application is configured to enable users to modify the properties of the motion like velocity and size and the typography like font

color and font style. To evaluate the methods of manufacturing language, we compare existing language named SVG and the primitive motion API. For this step, both applications contain typical motion which used in the kinetic typography postcard frequently.

5.2 Simplicity of the Source Code

First, elements of motion to express the motion such as text, path and time information chosen by users using SVG language need SVG file based on XML which has 2D vector graphics. However, the SVG language is hard to control relevant pixel information therefore developers should preset a path and movement along with the path. Table 3 shows SVG source code to generate straight motion and circular motion.

Table 3. Sample Source Code of SVG Language

Motion	Source code of SVG language
	<code>content+="<animate begin= \"0s \" dur= \"\"+Tab1. motion_speed+\" \"repeatCount= \"1 0\"attribute Name= \"startOffset \" values= \"100%;0% \" />"+\" </textPath> </text> <defs><path id= \"curve \" d= \"m27, 49.66c86, -0.22172,-0.44 258,1\"fill= \"none \" fill-opacity= \"0 \"stroke= \" #FFFFFF\" /> </defs>\";</code>
	<code>content+="<animate begin= \"0s \" dur= \"\"+Tab1. motion_speed+\" \" repeatCount= \"10 \" attributeName= \"startOffset \" values= \"100%;0% \" />"+\"</textPath> </text> <defs> <path id= \"curve \"d= \"m139, 111-16,11-18, 21-12.5, 0.51-12.5,0.51-19, 61-17,111-9,101-3, 1515,1516,8117,2135,1127,0132, 0119,0120, 0121,0115, 0124, -4112, -1016,-1210,-91-4, -91-9,-101-10,-61-15,-61-13,-11-17,-31-14,-11-9,01-12,11-15.5,-11-13.5,-1z\"fill= \"none\"fill-opacity= \"0\"stroke= \" #FFFFFF\" /> </defs>\";</code>

However, because primitive motion API generates text object and set properties of text and movement it is included in standard web environment simply. Users can use the API just calling the function with object and properties. Table 4 indicates source code of primitive motion API to construct straight motion and circular motion.

Table 4. Sample Source Code of Primitive Motion AMP



Motion	Source code of Primitive Motion API
	<code>kityLine(move,"right", 200, velocity);</code>
	<code>kityCircle(move, 360, 100, velocity);</code>

Figure 5 shows the result of comparing SVG source code with primitive motion API source code except the blanks. The primitive motion API source code indicates decrease magnification of 0.26 times compared with SVG source code and it proves convenient coding process performed by using primitive motion API.

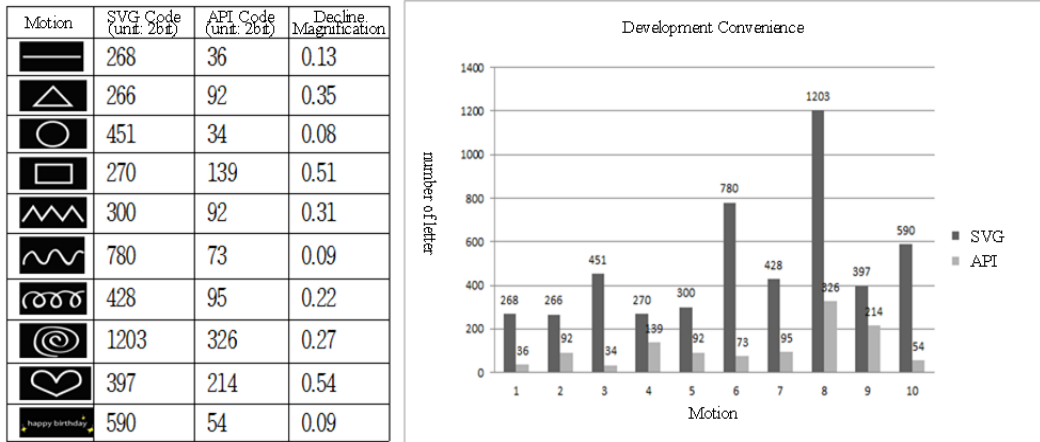


Figure 5. Development Convenience of Primitive Motion Library

5.3 Improvement of the Source Code Comprehension

We test the process of guessing the motion through source code of SVG and primitive motion API source codes to evaluate the source code comprehension. For this test, the experiment is confirmed by connecting the figure and the motion through the speculation process of source code.

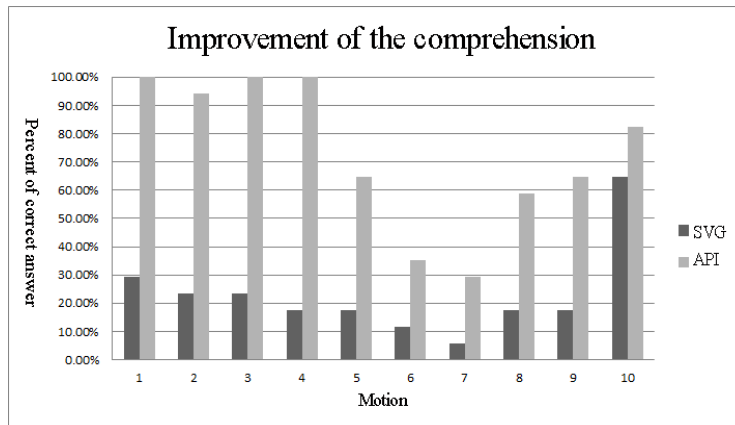


Figure 6. Improvement through Source Code

The percent of correct answer to match the very same motion with source code shows that SVG is 22.94% and primitive motion API is 72.94%. It indicates that primitive motion API is more obvious about 3.17 times than SVG does.

6. Conclusion

Kinetic typography indicates text movement by adding up the variety of effects of the moving text such as time, space, speed and weight to show both temporal and spatial moving. This paper constitutes primitive motion library using primitive motion API to substitute existing ways to manufacture the kinetic typography. For constructing the primitive motion library, we analyze the features of texts and properties of motion and reorganize it as primitive motion API to visualize kinetic typography effectively. Also, design

primitive motion library as format of standard web environment, as primitive motion API can provide users with a medium to yield particular kinetic typography conveniently and immediately in the social media environment. In addition, arranging the motion library for each domain can help the application build using kinetic typography. The reader should bear in mind that the study is based on primitive movement; accordingly, it should expand the range to express complicated motion. In other words, the combination of Primitive Motion API will be even advanced to diverse motions, promoting enrichment of users' emotions and kinetic typography.

Acknowledgements

This work was supported by the ICT R&D program of MSIP/IITP. [2014(KI0041794), Development of the Kinetic Typography Technology for Emotional Representations in Social Media Environment] and by Ministry of Culture, Sports and Tourism(MCST) and Korea Creative Content Agency(KOCCA) in the Culture Technology(CT) Research & Development Program 2014.

References

- [1] S. Ford, J. Forlizzi and S. Ishizaki, "Kinetic Typography: Issues in time-based presentation of text", ACM CHI 97, (1997).
- [2] Y. Y. Wong, "MIT, Master in 1995", presentation at CHI 96, (1996).
- [3] J. C. Lee and J. Forlizzi, "The kinetic typography engine: an extensible system for animating expressive text", Proceedings of the 15th annual ACM, (2002).
- [4] Z. Yeo and S. E. Hudson, "KTE2: An Engine for Kinetic Typography", CHI, (2009).
- [5] M. Kim, J. W. Jun, Y and C. Choi, "Partial Movement of Kinetic Typography Using Sketch-based Interfaces", KCC, (2013).
- [6] Y.-Y. Ha, S.-Y. Kim, I.-S. Park and S.-B. Lim, "Development of Video Caption Editor with Kinetic Typography", Multimedia Society, vol. 17, no. 3, (2014).
- [7] S.-B. Lim, H. Park and C. H. Lee, "HTML introduction of Web Programming", Hanbit Media, Korea, (2013).
- [8] J. E. Kwak and S.-B. Lim, "Design of Motion Library for using the Kinetic Typography in Messengers", KIISE, vol. 39, no. 22, (2012), pp. 190-191.
- [9] "Flash Professional", Adobe, <http://www.adobe.com>.
- [10] "After Effects", Adobe, <http://www.adobe.com>.

Authors



Yoon Ah Cho received her BS degree from SookMyung Women's University, Korea, in 2013. From 2013, she is currently a MS degree in Multimedia Science from SookMyung Women's University. Her main research interests are kinetic typography and user interface and smart service.



Joo Hyun Park received her BS and MS degree from SookMyung Women's University, Korea, in 2010 and in 2012.

From 2013, she is currently a PhD degree in Multimedia Science from SookMyung Women's University. Her main research interests are E-book, voice annotation, kinetic typography and user interface.



Soon-Bum Lim, he received his BS degree from Seoul National University , Korea, in 1982 and his MS and Ph.D degrees in computer science from KAIST(Korea Advanced Institute of Science and Technology), Korea, in 1983 and 1992, respectively. From 1989 to 1997, he was the engineering director for the font technology and printer division at Human Computer, Inc, and Trigem Computer, Inc. From 1997, he was an assistant professor in the Dept. of Computer Science at Konkuk University in Korea. From 2001, he is currently a professor in the Dept. of Multimedia Science at Sookmyung Women's University in Korea. His main research interests are computer graphics, web and mobile multimedia contents, user interface, and electronic publishing such as font, eBook, and XML documents.