Automated Physics Based Animation of Fonts

Nirmal Kumawat
Adobe Inc.
kumawat@adobe.com

ABSTRACT

We present a new method for automatic font animation, where the outlines of one or more glyphs are automatically segmented into sets of curves that can be efficiently controlled over time. This segmentation takes into account the directionality of similar segments, and allows arbitrary levels of subdivision. We showcase different examples of physics-based effects that can be driven by motion sensors in mobile devices to achieve novel and engaging text experiences. Our method works over existing off-the-shelf fonts, enabling new automatic text-based effects previously requiring lots of manual work.

CCS CONCEPTS

• **Computing methodologies → Computer graphics**.

KEYWORDS

Digital Typography, Font, Glyph, Vector Graphics

ACM Reference Format:
Nirmal Kumawat. 2020. Automated Physics Based Animation of Fonts. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Posters (SIGGRAPH '20 Posters), August 17, 2020.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3388770.3407434

1 INTRODUCTION AND MOTIVATION

Animation is a process of manipulating images to create moving images. In many cases, a user may wish to animate text. For instance, when making a video for personal or commercial reasons, the user may wish the video to include text that is animated in a way that is customized or grabs the interest of the viewer. For another example, a user may desire to use text animation as a form of expression for personal use, such as in text or email communication with friends or colleagues. Although some applications include tools that can be used to generate animations of text. These tools are difficult to use and sometimes require expertise on the part of the user. For instance, a user may have to convert text to an outline and then manually add points to the outline to enable text to be animated. Additionally, the user may be required to move those points manually to indicate how the text should change across keyframes during an animation.

2 TECHNICAL APPROACH

The font animation to be generated is based on a set of keyframes, each keyframe associated with a time. In each keyframe, each glyph point has a respective position and each segment of glyph outline has a respective appearance. In some examples, each glyph point can be treated as a point mass and thus, when a force is applied on glyph point, then font animation system moves glyph point according to laws of physics [Bargteil and Shinar 2019].

2.1 Conversion of Glyph Outline to Multiple Subcurves

The glyph outline (Bézier paths) contains both line and curve segments. The font animation system first converts each line segment to curve segment and then, each curve segment is subdivided into multiple subcurves with equal lengths, each of which is also a Bézier curve (Fig: 1).

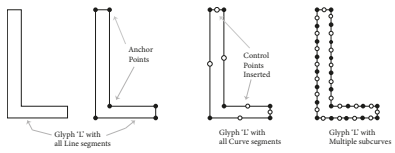

Figure 1: Bézier path is subdivided into multiple subcurves

2.2 Animation Keyframe Generation

Then the system determines the respective position of each glyph point at each time of the keyframes by applying an effect function such as, for instance, a force function that applies a force to reposition a glyph point or a velocity function that repositions the glyph points based on an indicated velocity (Fig: 2).

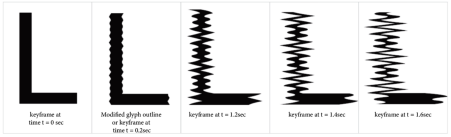

Figure 2: Each point $p(x, y)$ is modified with equation: $x(t) = x_0 + sin(y + t) * t^2$

In some examples, an animation effect is applied selectively to only a proper subset of the segments or glyph points, based on a direction of a segment. Direction of each segment is decided based on coordinates of its both end-points 3.

Fire Effect will be applied only on segments [figure: 3] with Left directions which are ([P12, P1], [P8, P9], [P4, P5]) as shown in figure [4].

Melt Effect will be applied on segments [figure: 3] with Right directions which are ([P10, P11], [P6, P7], [P2, P3]) as shown in figure [5]

SIGGRAPH '20 Posters, August 17, 2020, Virtual Event, USA

ACM ISBN 978-1-4503-7973-1/20/08.
https://doi.org/10.1145/3388770.3407434

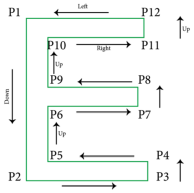

Figure 3: Segments with their directions

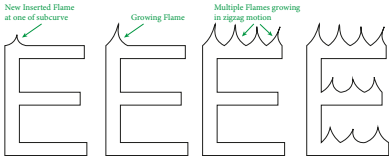

Figure 4: Fire effect generation based on directions

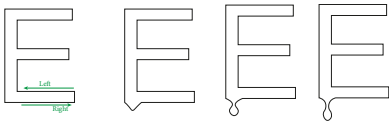

Figure 5: Melt effect generation based on directions

3 EXAMPLE EFFECTS

3.1 Wind Effect

Each point $p(x, y)$ of each subcurve will be modified as per equation $x(t) = x_0 + delta * W_X; y(t) = y_0 + delta * W_Y;$; where W_X, W_Y represent the wind speed in x and y directions respectively. $delta$ is computed with equation $delta = rand()\%F$; where $rand()$ generates random number between 1 and F; F is a factor which is (let's say) 5% of total height or width of glyph bbox (whichever is minimum). The wind speed can be simulated with physical device's motion sensor such as accelerometer.

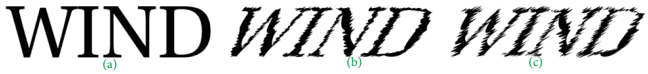

Figure 6: Intermediate keyframes of wind effect generated with device motion sensor

3.2 Hanging Cloth Effect

Each point $p(x, y)$ of each subcurve will be modified as per equation $x(t) = x_0 + k * sin((y - T)/H + t) * t + (y - T)/tan(angleRadian)$; $y(t) = y_0 + k * sin((xDiff)/W + t) * t$; where k is constant, T represents the top of glyph bbox (hanger point on rope) and H represents height of glyph bbox, W represents width of glyph bbox, $angleRadian$ represents angle value in x direction of device's state, $xDiff = MIN(x - glyphBoxXMin, glyphBoxXMax - x)$ Parameter t can be simulated with gyro-meter sensor data.

Figure 7: Intermediate keyframes of Hanging Cloth effect generated with device motion sensor

3.3 Fire Effect

A subcurve, based on its direction [fig: 4], will be replaced with flame outline (a Bézier curve) which will be growing as per time. Other subcurves will be modified as per zigzag motion to give realistic appearance. Here zigzag motion refers to random small variations in glyph points. A different kind of flame outline will generate different kind of Fire Effect, hence, a design application can provide multiple variants of Fire Effect.

Figure 8: Intermediate fire effect keyframes

3.4 Melt Effect

A subcurve, based on its direction [fig: 5], will be replaced with melt outline (a Bézier curve) which will be growing as per time. A different kind of melt outline will generate different kind of Melt Effect, hence, a design application can provide multiple variants of Melt Effect.

Figure 9: Intermediate melt effect keyframes

4 CONCLUSION

The effects mentioned in this poster paper are not just limited, the endless number of effects can be generated such as Water Flow, Wave, Rain etc. within few seconds by using device motion sensors such as accelerometer, gyroscope etc. on source device. Thus, it connects vector based digital fonts to real-world physics laws of motion.

REFERENCES

Adam W. Bargteil and Tamar Shinar. 2019. An Introduction to Physics-Based Animation. In *ACM SIGGRAPH 2019 Courses* (Los Angeles, California) *(SIGGRAPH '19)*. Association for Computing Machinery, New York, NY, USA, Article 2, 57 pages. https://doi.org/10.1145/3305366.3328050