HTML5 Graphic Animation technique: the remarkable difference between Canvas element and Scalable Vector Graphics (SVG) language

Sangyeol Kim (a1689938)
University of Adelaide, Australia
sangyeol.kim@student.adelaide.edu.au

***Abstract*—Recently, Internet has been an indispensable part of people's lives. Hence, numerous people have been using the Internet for various purposes such as shopping, study, entertainments, and etc. Naturally, the majority of web developers and organizations purses more advanced web technologies to reach Internet users' satisfaction and to increase the value of the offering. Web visualization rendering technique is a key part of the advanced web technology for their web project. Especially, a web animation of the technique is a vehicle for greatly faster click-through and higher arousal in a web project. Canvas and SVG are the most well-known HTML pure visualization methods, which support web developers to develop dynamic animations without a difficult. However, web beginner developers or computer science students might not specifically recognize the difference between Canvas and SVG Accordingly, this study will suggest a guideline for the comparison animation performance between Canvas and SVG with a benchmarking tool. Furthermore, the tool is running on the three popular web browsers and providing 90 CSV files, which include the FPS data of 300 times on a various test case. Consequently, this research will analyze the collected data and suggest a guideline for a web developer.**

I. Introduction

With the development of Information technology, the Internet users have been extremely increasing in all over the world [1]. They also have had different purposes of utilization of a web service. Generally, the majority of Internet users have been using a web service for shopping, study, entertainments, and etc[1]. Accordingly, numerous web developers and organizations have been applying more advanced web-native technology to their web service, in order to increase the value of offering and attract people's attention [1]. Typically, as they utilized a web visualization rendering techniques of a web-native technology, they helped Internet users to easily access visual information from their web service [2]. Canvas element and Scalable Vector Graphics (SVG) language are well-known methods to maximize visualization efficiency and resilience on a web [2] [3]. Hence, numerous developers may prefer to use the methods since Canvas and SVG are easy to be realized by JavaScript and the methods support a complex animation without a difficult. However, web beginner developers or computer science students might not specifically recognize the difference between Canvas and SVG because Canvas and SVG have different architectures for the same purpose. For instance, The Canvas allows for dynamic rendering of two-dimensional shapes with a bitmap image while SVG is an XML-based vector image format for two dimension graphics [2] [4]. Moreover, Canvas element is based on *immediate mode* that completely redraws a bitmap image on every frame initiated by JavaScript [2]. Whereas SVG is an language to express a vector image in XML that is based on *retained mode* [4]. Hence, Canvas could be showing better performance than SVG in a web animation. However, the animation performance must be different depending on the number and size of entities and different environments [5]. Therefore, this research will explore the different animation performance between Canvas and SVG and analyze the performance to establish a guideline for web beginner developers or computer science students.

A. Background

Since web2.0 has been introduced, web visualization technology has been contributed to a great revolution of Internet [6]. Furthermore, the performance of the web visualization technology can be similar to the performance of desktop application because of Rich Internet Application (RIAs) of Web2.0 [6]. Normally, the web-based visualization technology has been utilizing at different environments such as server-based and fully client-based environments [7]. Recently, client-based visualization technique has been preferred because the technique has been able to interactively communicate with a server page via "asynchronous server communication", which conducted by JavaScript [3]. Furthermore, the client-based visualization technique aims the reduction of the total cost of ownership (TCO) in server-client networks due to the principle of thin-client [8]. Hence, the visualization techniques' performance is influenced by client environment and each technique's architecture. Canvas is subordinated to HTML that aims a low-level procedural model, it supports a dynamic bitmap image on a web [2]. The basic concept of Canvas is to involve a two dimensions context in order to support a programmer to express various shapes, render, text, and display, which consists of bitmap image on a web browser window [2]. Furthermore, only a final image that generated by Canvas shall be stored in a memory due to a procedural rendering generation, it has to re-drawing by every request [3]. On the other hands, web developers used SVG which is a language to draw images on a web instead of Canvas element. Greatly, SVG supports three types of graphic objects: vector graphic shapes, images and text, which are similar to the graphics types of Canvas except for vector graphic type [4]. The vector graphics type consists of lines and curves as geometric objects, which is considerably flexible formats than a raster-based Canvas method [4]. Hence, web developer or computer science students are aware of that SVG shows a higher quality image than Canvas. Although the aforementioned Canvas and SVG technique have different architecture respectively, both techniques provide XML or JavaScript dialects in order to generate a dynamic web animation. A web animation is used for various purposes such as web game, web advertises, map, navigation, and etc [2]. Moreover, the majority of web developers and organizations utilized a web animation to increase the value of their web service. Absolutely, a web application with an animation is enough to attract Internet user's attention. Sundar and Kalyanaraman [5] mentioned that an animation is contributed in order to elicit greatly faster click-through and higher arousal. Therefore, the animation effect and responses on a web that could be strongly correlated to people's attention could be emphasized in the real world. Naturally, numerous developers and organizations will consider applying the visualization methods to their web project. Accordingly, this paper will figure out the different animation performance between Canvas and SVG with a benchmarking tool.

B. Goal and Expectation

To figure out the different animation performance between Canvas and SVG on a web, This paper developed two research questions as below :

- RQ1 : How do Canvas element differ from Scalable Vector Graphics (SVG) language for a web animation?
- RQ2 : How much a web animation by Canvas and SVG be affected depending on different web browsers?

According to the questions, this paper will discuss the comparison animation performance between Canvas and SVG on the most popular three browsers. Furthermore, a benchmarking tool will be given a series of sample renders to verify the animation speed of the visualization techniques. On the other hands, through the result data, the stability and graphics quality of Canvas and SVG animation will be identified. In this paper, Section 2 will give some literature reviews for the comparison between Canvas and SVG. Section 3 will show how to collect data and what is a benchmarking tool. Additionally, the analysis of collected data will be discussed in Section 4. Lastly, the summarizing of analysis of result data will be mentioned in Section 5.

II. Literature review

The research question and topic imply comparison web animation performance and advanced web visualization technology on the most popular three web browsers. Hence, this section will deal with greatly two categories: Web-native information visualization and Related works methodology in order to answer the research question and topic.

A. Web-native information visualization

Web 2.0 technologies emphasized a communication between web applications and users through the World Wide Web(WWW) [9]. To reach the principle of web 2.0, there are numerous technologies in web 2.0. Rich Internet Applications (RIAs) is a part of web 2.0 technology that aims for lightweight distribution architecture, high interactive interface and low computation time [10]. Typically, Adobe Flash, JavaFX and Microsoft Silverlight as web-native visualization technology in RIAs deal with dynamic web animation, drawing shapes and images to interact with users [10]. As mentioned the principle of RIAs, the performance of visualization technique shows a similar whether it is on a web or desktop. Reimers and Stewart [11] found that although the reaction time of Flash on laboratory computers illustrated a small difference with the reaction time of Flash on web-connected computers, there has been no significant difference performance between a web and desktop runtime environment. On the other hands, the visualization technique of RIAs shows better performance than HTML pure renderings such as Canvas and SVG. Johnson and Jankun-Kelly [3] concluded that a web-native visualization technique by Java applet had the best performance than Canvas and SVG. They also recommended that Java and Flash method are used in highly nested documents than HTML pure rendering because the pure rendering techniques are not optimized for the document [3]. Thus, web-based visualization methods in RIAs have a better performance than HTML pure rendering techniques. However, the majority of web developers avoid developing a Flash and Java animation of RIAs on a web due to the disadvantages of RIAs visualization platforms. Firstly, RIAs visualization platforms have to follow specific runtime environment. For instance, although Adobe Flash is a powerful visualization technique on a web, Internet users have to install a plug-in to their browser or computer, in order to use and see a Flash animation [11] [6]. The other is that RIAs visualization techniques require their own programming language to derive the best performance. Reimers and Stewart [11] mentioned that Flash and Java involve their complex programing language that could be showing different performance in across platform. Furthermore, a web programmer who deals with a front-end part in a web project has to spend extra time to learn and apply the complex programming language. Lastly, mobile web browsers do not support the Java and Flash animation because of security and performance problems in mobile web browsers. Hence, the large IT companies do not support Flash and Java animation plug-in in their mobile web browser [12]. The company point out that Flash is a closed system, it had the worst security record in 2009 and it contributes to wasting battery of mobile. The company point out that Flash is a closed system, it had the worst security record in 2009 and it contributes to wasting battery of mobile. Thus, RIAs visualization platforms had some disadvantages, which negatively impact to current IT industries. Therefore, this study focuses on Canvas and SVG, which are HTML pure visualization rendering techniques. Additionally, as comparing the two visualization techniques, this research will give a guideline for any web projects.

B. Related works methodology

To compare an animation performance between Canvas and SVG, visual results are significantly needed for this study. Likewise, other research that deal with the comparison performance between a web-native visualization techniques have also utilized appropriate methodologies to deduce results [3] [13] [14].

Johnson and Jankun-Kelly [3] explored the strengths and weaknesses of web-based visualization technologies. For their research purpose, they have used scalability studies with parallel coordinates and squarified treemaps, which showed numerous dataset as a graph on different runtime environments. Also, they found that HTML pure visualization technique (Canvas and SVG) have potential to expend the impact of web visualization area and they defined web-based visualization technique created by Java shows the best performance. Furthermore, Canvas showed a better performance than SVG in the complexity of document increased via their experiment. Consequently, they have used a rendering time with numerous dataset to define the performance of the visualization techniques. Kee, Salowitz and Chang [14] point out that web-based visualization techniques could show their performance as a rendering speed. Additionally, they identified the interactive performance of Canvas and SVG with 2D and WebGL library using mouseover testing. Although they illustrated a possible support data with a rendering speed testing, the result of the research was significantly ambiguous. They combined web visualization techniques to figure out the best web rendering methods and they tested the combined techniques to determine the rendering speed and performance. As a result, they did not have a specific support data and they avoided to mention a single rendering performance. Thus, to compare a web-based visualization technique, other research utilized rendering time and speed on different runtime environments. Hence, this research will deal with the rendering performance as an animation performance on different browsers. However, in this paper, a benchmarking tool measure only Frame per seconds (FPS) to estimate the animation performance of Canvas and SVG because the animation speed implies rendering time and speed at the same time. Additionally, to measure FPS value of Canvas and SVG, the benchmarking tool render a series of particles first and calculate FPS values.

III. Methodology

The comparison animation performance between Canvas and SVG required the visible result to judge which technique show a better performance on specific conditions. Accordingly, a benchmarking tool is developed to collect data from Canvas and SVG on the three web browsers. In this section, the research will demonstrate how to collect data and what kind of data the study needed.

A. Experimental Setup

*1) **Hardware Configurations**:* Canvas and SVG are client-based visualization techniques in client-server networking.

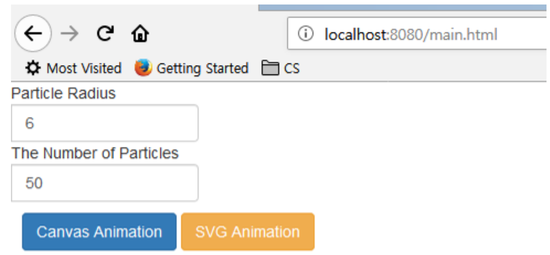

Figure 1. the user interface of a benchmarking tool

Hence, the performance of the techniques could be affected by client hardware performance. For several testings, identical hardware configurations are set up as below :

- Display : 1366 x 768
- OS : Windows 7 Professional K 64 bit
- CPU : i7-2670 2.2 Ghz
- RAM : 8 GB
- Web browsers
  - Chrome 65.0 64 bit (Blink)
  - Firefox 59.0.2 64 bit (Gecko)
  - Internet Explorer 11.0.96 64 bit (Trident)

The benchmarking tool was running on a personal laptop, which includes the hardware configurations. Additionally, The experimental setup involves three web browsers because each browser has different rendering engine.

*2) **Benchmarking Tool**:* The benchmarking tool [1] that was used to compare the animation performance of Canvas and SVG was developed using a base-source, which came from "*The Man In Blue*" [2]. The benchmarking tool consists of HTML5, JavaScript and Bootstrap[3], in order to test animation performance of Canvas and SVG. The tool also has a simple user interface that created by Bootstrap (Fig. 1) to control the number and size of the particle and the type of web visualization techniques. On the other hands, when the benchmarking is running with Canvas or SVG, the tool calculates the frame per second according to the number and size of particles. Additionally, the benchmarking tool collected the FPS data of 300 times and give CSV file, which stored FPS data.

*3) **Local Server environment**:* Canvas and SVG are HTML pure visualization techniques that need a simple server-client environment to measure the performance as a client-based visualization technique. Hence, a local server environment was established for running a benchmarking tool. The local server conducted by Spring Boot[4]. The Spring Boot provides

[1]Canvas-SVG-Benchmarking tool : https://github.com/harry1913/Canvas-SVG-Benchmarking

[2]The Man in Blue : http://themaninblue.com/

[3]Bootstrap : https://getbootstrap.com/

[4]Spring Boot : https://spring.io/projects/spring-boot

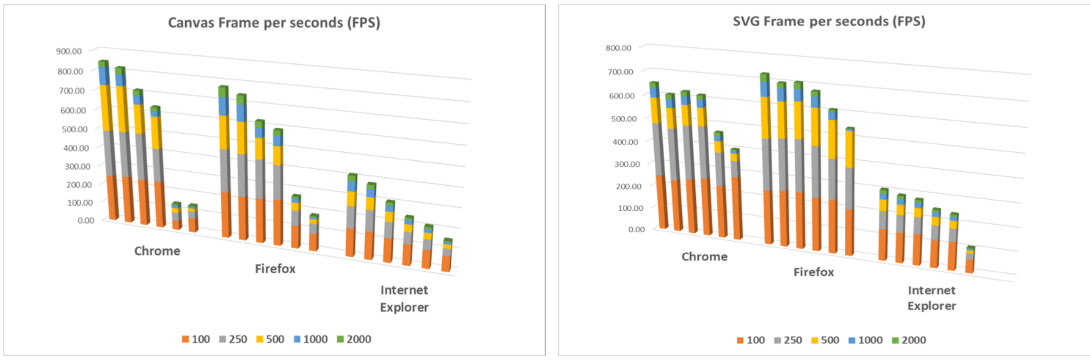

Figure 2. The trend of FPS per the number and size of particles on the three web browsers

simple configurations and tools, which manage dependencies to easily operate a web server. Furthermore, the Spring Boot support HTML5, CSS3, JavaScript, Bootstrap, SVG, and etc, in order to build a server-client environment.

*4) **Different Web browsers**:* The experimental setup involves three different web browsers to implement a benchmarking tool because the three web browsers have different architectures and rendering engines. Especially, the rendering engines can largely contribute to animation performance of Canvas and SVG because a rendering engine in a web browser is concerned in supporting HTML, JavaScript and CSS on a browser window[15]. This study selected three web browsers: Chrome, Firefox and Internet Explorer that are the most popular web browsers. In addition, each web browser has specific rendering engine as blow :

- *Blink* is Chrome web browser rendering engine. Since Chrome 28.0, the rendering engine of Chrome was changed from *WebKit* to *Blink*. *Blink* is divined from *WebKit* engine that contribute to optimizing process as a part of multi process application in Chrome [16].
- *Gecko* is Firefox web browser rendering engine. *Gecko* analyzes core functions of a rendering engine such as HTML, CSS, XUL, and JavaScript and then present constructed core function on a browser screen [17].
- *Trident* is Internet Explorer web browser rendering engine that is the oldest rendering engine. Furthermore, the engine has not been updated a longtime because it was exclusive rendering engine [18]. Hence, the scalability, compatibility and performance of *Trident* are poor than *Blink* and *Gecko*.

Thus, there are three different web browsers and rendering engines, which may be related to the animation performance of Canvas and SVG.

B. Data Collection

The benchmarking tool provides FPS values to evaluate the animation speed of Canvas and SVG. Also, the tool was running with different variables on the three web browsers as below :

- Chrome
  - Canvas
    - A series of particle Radius (1, 6, 24, 48, 150, 300)
    - The number of particles (100, 250, 500, 1000, 2000)
  - SVG
- Firefox
  - Canvas
    - A series of particle Radius (1, 6, 24, 48, 150, 300)
    - The number of particles (100, 250, 500, 1000, 2000)
  - SVG
- Explore
  - Canvas
    - A series of particle Radius (1, 6, 24, 48, 150, 300)
    - The number of particles (100, 250, 500, 1000, 2000)
  - SVG

A series of particle radius (1, 6, 24, 48, 150, 300) presented the size of each particle that could be related to that how the different size of particles affected the animation performance. Especially, the radius of less 1 and over 300 might be limited to use at an animation. Because the benchmarking tool showed maximum FPS rate at the radius of 1 and minimum FPS rate at the radius of 300. The other is the range of the number of particles (100, 250, 500, 1000, 2000) that also may have an effect on the animation performance of Canvas and SVG. Thus, this study collected FPS rates by the series of particles radius and quantity from a benchmarking tool. Furthermore, the benchmarking tool collects FPS 300 times and provide CSV file, which stored the FPS data to calculate average FPS data. Consequently, each web browser has 30 CSV files, total 90 CSV files are used to compare the animation performance of Canvas and SVG on different web browsers.

C. Animation Quality

The animation quality may have two aspects: Graphics quality and stability quality. The graphics quality of Canvas and SVG was evaluated from a subjective point of view with sample images, which came from a benchmarking tool. Additionally, by using a web browser zoom function, the simple image will be presented in five times zoom, in order to identify the graphics quality. On the other hands, the stability of Canvas

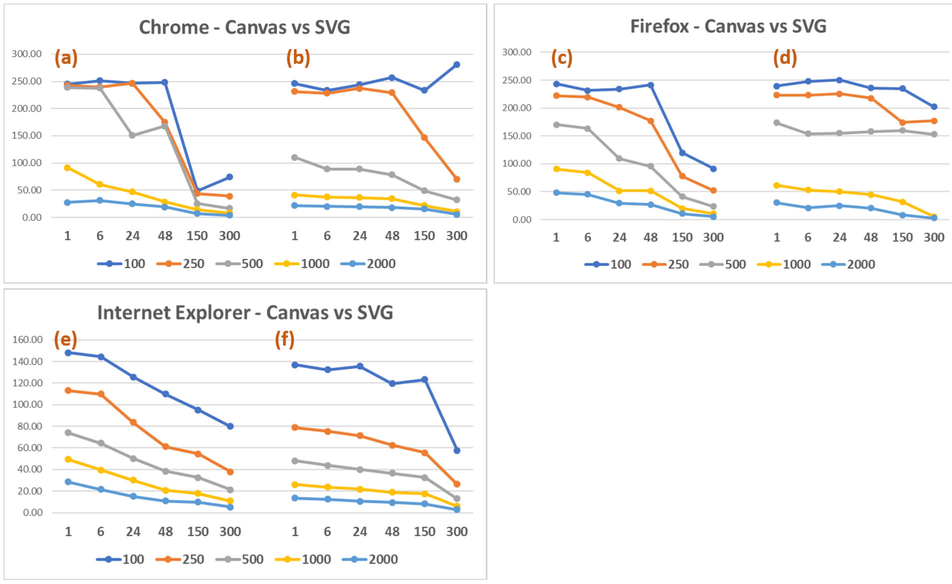

Figure 3. Canvas VS SVG on three web browsers

and SVG animation was defined from the 300 times of FPS data of 2000 quantities of particles.

IV. Analysis

The collected data included FPS data, which depend on the number and size of particles on each web browser, it will show the trend of animation performance of Canvas and SVG and it will present the comparison animation performance between Canvas and SVG. Lastly, sample images and the stability of animation performance will illustrate the quality of animation for Canvas and SVG depending on each web browser.

A. Animation Performance with FPS

*1) **The trend FPS of Canvas and SVG on different web browsers**:* Fig. 2 shows total FPS data per the different size of particles on different web browsers. In the Canvas FPS, it shows better performance on Chrome and Firefox than Internet Explorer expectedly. However, Canvas on the 150 and 300 size of testing presents a similar performance on all browsers. On the range of 150 and 300, Firefox shows the best performance than others and Internet Explorer is a next better performance and Chrome shows the lack of performance at the range. However, in the SVG FPS, Internet Explorer shows the poorest FPS performance while Firefox has the best FPS performance for the all scale of particles. Although Internet Explorer has the oldest rendering engine for the HTML pure rendering techniques, in the large size of particles, Canvas on Internet Explorer had a better FPS rate than Chrome. However, in the SVG testing, Internet Explorer has the poorest FPS performance between all web browsers.

*2) **Canvas vs SVG on each web browser**:* On the other hands, the FPS values of Canvas and SVG indicates a different output on each web browser. As shown in Fig. 3, the result of Canvas on Chrome of (a) seems to be similar performance to SVG on Chrome of (b), but from the scale of 150, the FPS of (a) has rapidly decreased whereas the FPS of (b) shows a little increase FPS values for the 100 quantities of particles. The other is that the result of Canvas on Firefox of (c) illustrates steady decreasing depending on the number and size of particles while SVG on Firefox of (d) show a higher FPS value than Canvas of (c) on the same particles. The SVG on Firefox of (d) shows the best performance on the small numbers of particles which come from 100 to 500 than others. Lastly, Canvas (e) and SVG (f) on Internet Explorer presents a similar trend, which is constantly decreased depending on the number and size of particles. Overall, Canvas (e) shows a little better speeds than SVG (f) on all testing cases. Thus, the animation speed of Canvas and SVG has been

Table I
THE DIFFERENCE GAP BETWEEN CANVAS AND SVG ON THE TEST CASE OF PARTICLE SIZES

	1	6	24	48	150	300
Chrome	12.97%	14.87%	6.63%	1.71%	53.93%	47.60%
Firefox	3.06%	3.16%	6.03%	6.70%	38.61%	49.43%
Internet Explorer	15.32%	13.82%	4.36%	1.21%	6.12%	18.66%
Average	10.42%	10.62%	5.67%	3.21%	32.89%	38.56%

Table II
THE DIFFERENCE GAP BETWEEN CANVAS AND SVG ON THE TEST CASE OF PARTICLE NUMBERS

	100	250	500	1000	2000
Chrome	14.60%	7.32%	30.31%	15.63%	6.05%
Firefox	9.69%	13.26%	22.44%	11.12%	20.81%
Internet Explorer	0.15%	10.85%	13.61%	19.08%	22.65%
Average	8.15%	10.48%	22.12%	15.28%	16.50%

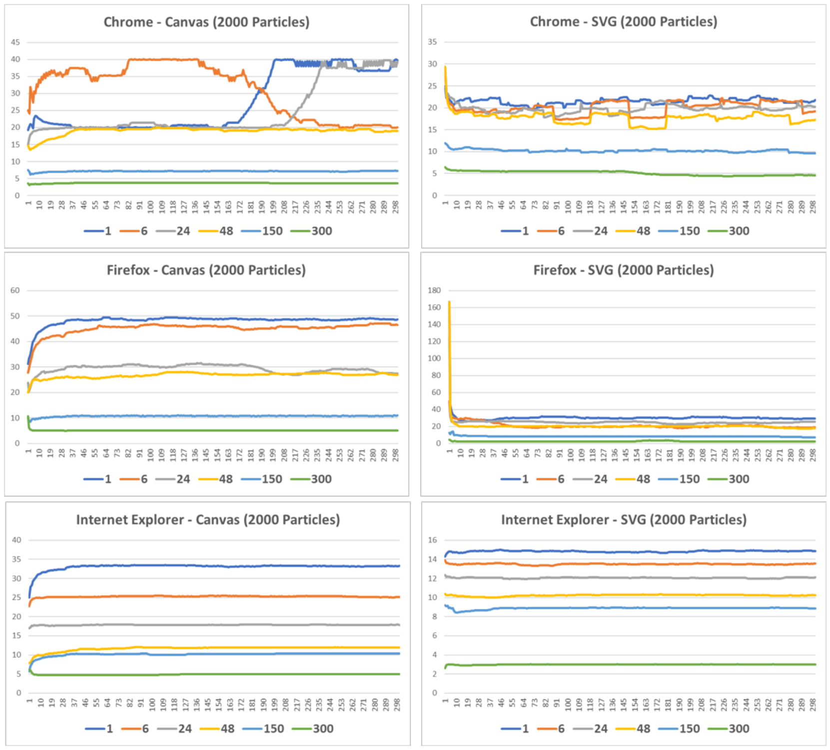

Figure 4. The project focuses of three professional developers

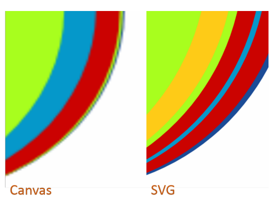

Figure 5. Graphic quality in 5 times zoom

affected by the number and size of particles. As shown in Fig. 3, Canvas has a better speed than SVG on the small scale testing case while SVG shows a little higher FPS rates on the large-scale testing case except for Internet Explorer.

*3) **the different gaps between Canvas and SVG**:* Table I shows how much Canvas and SVG differ depending on the size of particles. Especially, the large scale of testing has a large gap between Canvas and SVG. In the case of 1 size testing, approximately 10.45 % difference of gaps is presented while around 38.56 % difference of gaps on all web browsers is showed. The cases of 150 and 300 size testing had a large difference gap between Chrome and Firefox whereas Internet Explorer had more stable gaps than others. The case of 48 size testing presented the smallest difference gaps average 3.21 % while average 38.56 % is shown in the case of 300 testing. On the other hands, Table II shows the difference gaps of Canvas and SVG depending on the number of particles. The difference gaps in the case of 500 quantities testing on Chrome shows the largest difference gap between Canvas and SVG. Likewise, the case of 500 quantities testing on all web browsers indicate a large difference gap as average 22.12 %.

B. *Quality of Animation*

As mentioned above, Canvas is raster-based shape rendering technique while SVG is vector-based shape rendering method on a web [2] [4]. Hence, the image quality of SVG is extremely better. Fig. 5 shows a particle's graphics quality in five times zoom. A particle that is created by Canvas has a strong blur at the edge whereas a particles that is generated by SVG language shows high-quality graphic image. The result is expected without any doubts because of the difference based architecture. Also, this study found that the stability of animation speed is also a part of quality of animation on a web. As shown in Fig. 4, there are the 300 times collected FPS data of 2000 particles, which indicate a stability FPS rate on each web browser and the test case of 2000 particles means the worst scenario for the animation speed that is a reason why the test case was selected, in order to identify the stability of Canvas and SVG. One of interesting is that Internet Explorer shows the most stability on the conditions whereas Chrome shows an unstable FPS flow. Additionally, SVG language presented more stable than Canvas element on all web browsers.

V. Conclusion

A benchmarking tool provides numerous FPS rate data to answer the research questions, it also shows the trend of different animation performance between Canvas and SVG on the most popular web browsers. The criteria of a benchmarking have two perspectives: the number of particles and the size of particles. Through the analysis of the FPS rate data, this study found that the animation of Canvas shows a better performance at the small size and number of particles whereas the animation of SVG has a positive effect on the large scale of particles in each web browser. Furthermore, there is a large different gap in the largest size (300) of particles and the test case of 500 quantities show the largest different gap between Canvas and SVG. On the other hands, each web browsers has it own rendering engine, which contributes to animation performance. Typically, Chrome and Firefox involved more an advanced rendering engine than Internet Explorer. Hence, absolutely, the benchmarking data with Chrome and Firefox have high FPS rate data than Internet Explorer. However, all testing with the number and size of particles on Internet Explorer indicated the lowest difference gap between Canvas and SVG than other web browsers. Additionally, the result data on Internet Explorer showed more stability of FPS rates than other web browsers through the benchmarking tool and Firefox can derive considerably the high animation speed of SVG than other browsers. In the Canvas animation test, Chrome and Firefox showed a similar speed on the size and number of particles. The benchmarking tool draws the number of circle particles conducted by Canvas and SVG. On the browser screen, obviously, SVG offers higher rendering quality than Canvas because of the different architectures with Canvas. Furthermore, the SVG demonstrated more smooth animation for a testing. Consequently, Canvas and SVG are the best techniques to support web visualization effect on web browsers and its animation is sufficiently enough to attract Internet user's attention. Additionally, Canvas and SVG animation show different results depending on the scale of entities and browser. Through the result of this study, Canvas will be a suitable as a web project, which requires the number of small size animation object. Whereas, SVG will be fitted to a web project, which used a large size animation entity.

References

[1] M. Meeker, "Internet trends 2015-code conference," *Glokalde*, vol. 1, no. 3, 2015.

[2] S. Fulton and J. Fulton, *HTML5 canvas: native interactivity and animation for the web*. Ö'Reilly Media, Inc.", 2013.

[3] D. W. Johnson and T. Jankun-Kelly, "A scalability study of web-native information visualization," in *Proceedings of graphics interface 2008*. Canadian Information Processing Society, 2008, pp. 163–168.

[4] J. Ferraiolo, F. Jun, and D. Jackson, *Scalable vector graphics (SVG) 1.0 specification*. iuniverse, 2000.

[5] S. S. Sundar and S. Kalyanaraman, "Arousal, memory, and impression-formation effects of animation speed in web advertising," *Journal of Advertising*, vol. 33, no. 1, pp. 7–17, 2004.

[6] X. Gong, Y. Jin, Y. Cui, and T. Yang, "Web visualization of distributed network measurement system based on html5," in *Cloud Computing and Intelligent Systems (CCIS), 2012 IEEE 2nd International Conference on*, vol. 2. IEEE, 2012, pp. 519–523.

[7] J. Wood, K. Brodlie, and H. Wright, "Visualization over the world wide web and its application to environmental data," in *Proceedings of the 7th conference on Visualization'96*. IEEE Computer Society Press, 1996, pp. 81–ff.

[8] J. P. Kanter, "Understanding thin-client/server computing," 1998.

[9] S. Aghaei, M. A. Nematbakhsh, and H. K. Farsani, "Evolution of the world wide web: From web 1.0 to web 4.0," *International Journal of Web & Semantic Technology*, vol. 3, no. 1, p. 1, 2012.

[10] P. Fraternali, G. Rossi, and F. Sánchez-Figueroa, "Rich internet applications," *IEEE Internet Computing*, vol. 14, no. 3, pp. 9–12, 2010.

[11] S. Reimers and N. Stewart, "Adobe flash as a medium for online experimentation: A test of reaction time measurement capabilities," *Behavior Research Methods*, vol. 39, no. 3, pp. 365–370, 2007.

[12] S. Jobs, "Thoughts on flash," *Apple, Inc*, 2010.

[13] P. Corcoran, P. Mooney, A. C. Winstanley, and M. Bertolotto, "Effective vector data transmission and visualization using html5," 2011.

[14] D. E. Kee, L. Salowitz, and R. Chang, "Comparing interactive web-based visualization rendering techniques," *Tufts University, Medford, MA*, 2012.

[15] Wikipedia, "Web browser engine," http://en.wikipedia.org/w/index.php?title=Web%20browser%20engine&oldid=844907973, 2018, [Online; accessed 08-June-2018].

[16] ——, "Blink (web engine) — Wikipedia, the free encyclopedia," http://en.wikipedia.org/w/index.php?title=Blink%20(web%20engine)&oldid=844480839, 2018, [Online; accessed 09-June-2018].

[17] ——, "Gecko (software)," http://en.wikipedia.org/w/index.php?title=Gecko%20(software)&oldid=844249336, 2018, [Online; accessed 08-June-2018].

[18] ——, "Trident (software) — Wikipedia, the free encyclopedia," http://en.wikipedia.org/w/index.php?title=Trident%20(software)&oldid=843837697, 2018, [Online; accessed 09-June-2018].