# Prototype of MixVRT Which Is a Visual Regression Testing Tool That Highlights Layout Defects in Web Pages

**Naoki Aridome*, Tetsuro Katayama*, Yoshihiro Kita†,**
**Hisaaki Yamaba*, Kentaro Aburada*, and Naonobu Okazaki***
** Department of Computer Science and Systems Engineering, Faculty of Engineering, University of Miyazaki*
*1-1 Gakuen-kibanadai nishi, Miyazaki, 889-2192 Japan*
*†Department of Information Security, Faculty of Information Systems, Siebold Campus, University of Nagasaki*
*1-1-1 Manabino, Nagayo-cho, Nishi-Sonogi-gun, Nagasaki, 851-2195 Japan*
*E-mail: aridome@earth.cs.miyazaki-u.ac.jp, kat@cs.miyazaki-u.ac.jp, kita@sun.ac.jp,*
*yamaba@cs.miyazaki-u.ac.jp, aburada@cs.miyazaki-u.ac.jp, oka@cs.miyazaki-u.ac.jp*

**Abstract**

As a method for detecting layout defects in web pages, image-based visual regression testing is proposed. However, it has the problem that it takes time to detect unintended layout differences that are not based on HTML code changes. This paper proposes a prototype of MixVRT which is a tool to detect layout defects in web pages. It is a visual regression testing tool that highlights layout defects in web pages. From evaluation experiments, the time find to layout defects can be reduced.

*Keywords*: MixVRT, web page, layout defects, visual regression testing, HTML code

## 1. Introduction

In recent years, web pages need to be updated more frequently from the perspective of user experience (UX) and search engine optimization (SEO) [1]. Here, in changing to update a web page, there is a problem that intended layout differences that are based on HTML code changes and unintended layout differences that are not based on HTML code changes are mixed, and it takes time to distinguish between the two. In this paper, differences in layout unintended by the developer are referred to as layout defects.

To solve the problem, there is visual regression testing [2]. In the visual regression testing for web pages, the web pages before and after changes are placed side by side and differences in layout are highlighted to make it easier to spot differences in layout. Many existing studies of visual regression testing compare only images before and after changes and highlight all layout differences, including differences in the developer's intended layout [3][4].

However, finding layout defects is time-consuming because it requires comparing the differences between all the displayed layouts and the HTML codes to determine whether the differences are in line with the developer's intent. Therefore, to reduce the time to find layout defects in web pages, this paper develops a prototype of MixVRT which is a visual regression testing tool that highlights layout defects in web pages.

## 2. Functions of MixVRT

In this chapter, we describe the functions of MixVRT. It detects differences by comparing images and detects changes due to HTML codes changes on the web pages before and after changes and compares them to detect layout defects. In addition, it highlights the following three areas by surrounding them with red or green borders.

- Difference Areas
  Difference Areas is the areas removed from the web page before the change and the areas added to the web page after the change by comparing the images of the web pages before and after changes.
- Modified Areas
  Modified Areas is the areas of screen elements affected by changes in either the body element or the style element, or both, of the HTML code. These changes are detected by comparing the HTML codes of the web pages before and after changes. In this paper, Modified Areas is the areas of the intended layout differences.

*© The 2025 International Conference on Artificial Life and Robotics (ICAROB2025), Feb.13-16, J:COM HorutoHall, Oita, Japan*

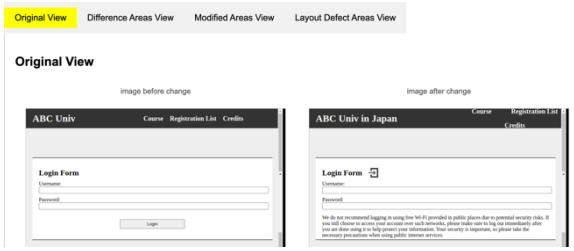


Fig. 1. Appearance of Original View

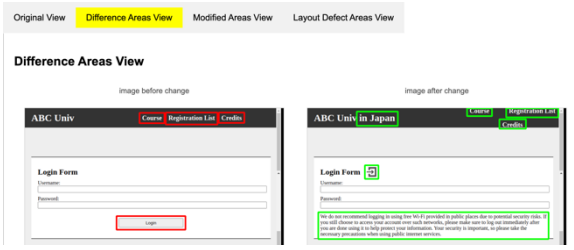


Fig. 2. Appearance of Difference Areas View

- Layout Defect Areas
  Layout Defect Areas is the areas of layout differences excluding the Modified Areas from the Difference Areas. In this paper, Layout Defect Areas is the areas of the unintended layout differences. In addition, Layout Defect Areas shows areas where it is possible that layout defects occur.

## 3. Appearance and Structure of MixVRT

In this chapter, we describe the appearance and structure of MixVRT. It takes as input URLs of the web page before and after changes. Then, for each of the four views shown below that make up the appearance of MixVRT, it generates a PNG format image corresponding to each view and outputs the image to a web page running on the local server for display.

- Original View
  The before and after images of the web page are displayed side by side on the left and right. The appearance of Original View is shown in Fig. 1.
- Difference Areas View
  The before and after images of the web page are displayed side by side on the left and right, with the Difference Areas highlighted with red or green borders. The appearance of Difference Area View is shown in Fig. 2.
- Modified Areas View
  The before and after images of the web page are displayed side by side on the left and right, with the Modified Areas highlighted with red or green borders. The appearance of Modified Area View is shown in Fig. 3.

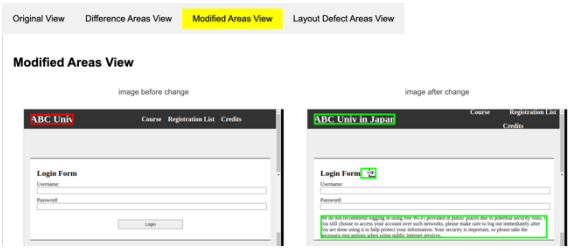


Fig. 3. Appearance of Modified Areas View

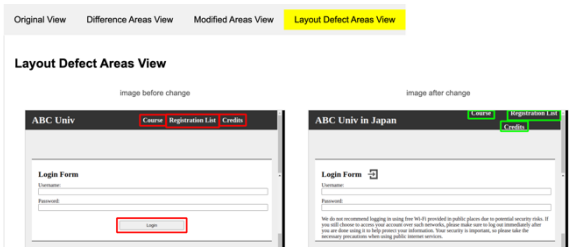


Fig. 4. Appearance of Layout Defect Areas View

- Layout Defect Areas View
  The before and after images of the web page are displayed side by side on the left and right, with the Layout Defect Areas highlighted with red or green borders. The appearance of Layout Defect Areas View is shown in Fig. 4.

The structure of MixVRT is shown in Fig. 5. MixVRT consists of the following five processing sections.

- Data Manager
  Data Manager is responsible for communicating data between each processing section, exchanging data with other processing sections, and maintaining data. The data in this paper are the images and HTML codes of the web pages before and after processing.
- Data Getter
  Data Getter acquires the images of the web page from the URLs of the web page received as input using Selenium WebDriver [5]. It also acquires the HTML codes of the web page using Requests [6]. These processes are performed for both the URL of the web page before and after changes to obtain the before and after changes images of the web page. The acquired data is output to the Data Manager.

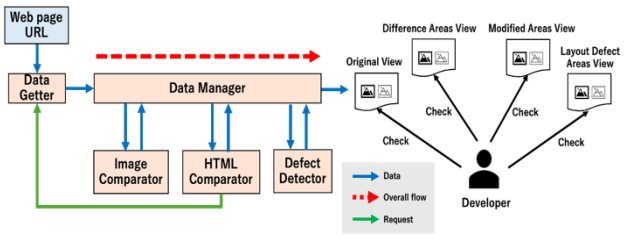


Fig. 5. Structure of MixVRT

- Image Comparator
  Image Comparator compares the before and after changes images of the web page received from the Data Manager in pixel units and generates the before and after changes images of the web page, highlighting the Difference Areas. The generated images are output to the Data Manager.
- HTML Comparator
  HTML Comparator compares the before and after changes HTML codes of the web page received from the Data Manager line by line using difflib [7] and generates a difference code. Then, based on the difference code, it adds HTML codes that encloses the changed HTML elements with red or green borders to both the before and after changes HTML codes. The added before and after HTML codes are then rendered as web pages, and by capturing images of the rendered web pages, the before and after changes images of the web page, highlighting the Modified Areas, are obtained. The obtained images are output to the Data Manager.
- Defect Detector
  Defect Detector detects Layout Defect Areas by comparing the images highlighting the Difference Areas with the images highlighting the Modified Areas and generates the before and after changes images of the web page, highlighting the Layout Defect Areas. The generated images are output to the Data Manager. The algorithm for detecting Layout Defect Areas compares the red or green borders of the Difference Areas and Modified Areas, respectively, and if the overlapping areas of the borders is less than 50%, it is determined to be Layout Defect Areas.

## 4. Evaluation of MixVRT

In this chapter, to evaluate the usefulness of MixVRT, we experiment to compare the time to find Layout Defect Areas and the detection accuracy of Layout Defect Areas between using MixVRT and conventional methods of image-based visual regression testing. The subjects are four students of University of Miyazaki.

In preparation for the experiment, we first prepare two web pages. Let these be Web page α and Web page β before the change, respectively. Web page α consists of 261 lines of code and 17 CSS classes, while Web page β consists of 472 lines of code and 28 CSS classes. Next, we change each of these web pages, each containing three Layout Defect Areas with layout defects. After the changes, Web page α consists of 275 lines of code and 20 CSS classes, and Web page β consists of 489 lines of code and 31 CSS classes. Let these be Web page α and Web page β after the change, respectively. In addition, we prepare an experimental file for the subjects to record the Layout Defect Areas when they found it.

### 4.1 *Experimental Methods*

We experiment using Web page α before and after changes, and Web page β before and after changes, under the following two cases:

- Case A: Using MixVRT, we measure the time to find Layout Defect Areas. It is the time from when a subject views the Layout Defect Areas View to when all Layout Defect Areas are found.
- Case B: Using conventional methods, we measure the time to find Layout Defect Areas. It is the time from when a subject views the Difference Areas View to when all Layout Defect Areas are found.

Here, the subjects can view the HTML codes before and after changes of the Web page for the experiment.

### 4.2 *Evaluation of time to find Layout Defect Areas*

Table 1 shows the time to find Layout Defect Areas in the two cases. From Table 1, it reduced an average of 12 m 59.5 s (about 91.8%) of time in Web page α and reduced an average of 33 m 41.5 s (about 97.9%) of time in Web page β. Here, we considered the reason why the reduction rate is higher for Web page β than for Web page α. This is because Web page β has more lines of code and CSS classes than Web page α (see Section 4), which increases the time to check the HTML codes. As a result, using the conventional methods takes significantly longer to find Layout Defect Areas compared to MixVRT. From the above, MixVRT can reduce the time to verify whether layout defects are in the areas where layout changes have occurred by checking the HTML codes.

Table 1. The time to find Layout Defect Areas

| | **Web page α** | | **Web page β** | |
|---|---|---|---|---|
| **Subjects** | **Case A** | **Case B** | **Case A** | **Case B** |
| 1 | - | 18m 41s | 52s | - |
| 2 | - | 9m 38s | 45s | - |
| 3 | 1m 36s | - | - | 32m 43s |
| 4 | 44s | - | - | 36m 7s |
| **Average** | 1m 10s | 14m 9.5s | 48.5s | 34m 25s |
| **Diff** | **12m 59.5s** | | **33m 41.5s** | |

### 4.3 *Evaluation of detection accuracy of Layout Defect Areas*

Table 2 shows the number of Layout Defect Areas over-detected and under-detected in Case B. In evaluation experiments, the subjects sometimes over-detect or under-detect Layout Defect Areas using the conventional method. In contrast, by using MixVRT, the subjects were able to detect Layout Defect Areas without over- or under-detection. In summary, MixVRT is more useful for detecting layout defects than the conventional methods of image-based visual regression testing.

Table 2. The number of Layout Defect Areas over-detected and under-detected in Case B

| Web page | Layout Defect Areas | Subjects | Over-detected | Under-detected |
|---|---|---|---|---|
| $\alpha$ | 3 | 1 | 1 | 1 |
| | | 2 | 0 | 1 |
| $\beta$ | 3 | 3 | 0 | 0 |
| | | 4 | 0 | 1 |

## 5. Conclusion

In this paper, we have developed a prototype of MixVRT which is a visual regression testing tool that highlights layout defects in web pages, to reduce the time to find layout defects in web pages.

We have experimented with two Web page. As a result, MixVRT can reduce the time by 91.8% and 97.9%. We also have confirmed that MixVRT detects all Layout Defect Areas without over- or under-detection. On the other hand, in the conventional methods, Layout Defect Areas have been over-detected or under-detected. Consequently, MixVRT is useful in reducing the time to find layout defects compared to the conventional method.

The following are future issues in this paper.

- Support for multiple browsers
- Support for different window sizes before and after web page changes

## References


1. CAES Office of Information Technology: “Why is web content so important?”, https://oit.caes.uga.edu/why-is-web-content-so-important/. (Accessed 2024-12-10)
2. Visual regression testing: awesome-regression-testing, https://github.com/mojoaxel/awesome-regression-testing. (Accessed 2024-12-10)
3. Haruto Tanno, Yu Adachi, Yu Yoshimura, et al.: “Region-based Detection of Essential Differences in Image-based Visual Regression Testing”, Journal of Information Processing, Vol.28, pp.268-278, 2020.
4. Akihiro Tsukakoshi: “Empirical Evaluation of a Visual Regression Testing Difference Detection Method Using Hierarchical Structures of GUI Elements” (in Japanese), Software Quality Profession, Vol.5, pp.207-214, 2020.
5. Selenium WebDriver: “WebDriver”, Selenium Documentation, https://www.selenium.dev/ja/documentation/webdriver/. (Accessed 2024-12-10)
6. PyPI: “Requests”, https://pypi.org/project/requests/. (Accessed 2024-12-10)
7. Python Software Foundation: “difflib − Helper for computing deltas”, Python 3.13.1 Documentation, https://docs.python.org/ja/3/library/difflib.html. (Accessed 2024-12-10)


## Authors Introduction


Tetsuro Katayama

Tetsuro Katayama received a Ph.D. degree in engineering from Kyushu University, Fukuoka, Japan, in 1996. From 1996 to 2000, he has been a Research Associate at the Graduate School of Information Science, Nara Institute of Science and Technology, Japan. Since 2000 he has been an Associate Professor at the Faculty of Engineering, Miyazaki University, Japan. He is currently a Professor with the Faculty of Engineering, University of Miyazaki, Japan. His research interests include software testing and quality. He is a member of the IPSJ, IEICE, and JSSST.

Yoshihiro Kita

Yoshihiro Kita received a Ph.D. degree in systems engineering from the University of Miyazaki, Japan, in 2011. He is currently an Associate Professor with the Faculty of Information Systems, University of Nagasaki, Japan. His research interests include software testing and biometrics authentication.

Hisaaki Yamaba

Hisaaki Yamaba received the B.S. and M.S. degrees in chemical engineering from the Tokyo Institute of Technology, Japan, in 1988 and 1990, respectively, and the Ph D. degree in systems engineering from the University of Miyazaki, Japan in 2011. He is currently an Assistant Professor with the Faculty of Engineering, University of Miyazaki, Japan. His research interests include network security and user authentication. He is a member of SICE and SCEJ.

Kentaro Aburada

Kentaro Aburada received the B.S., M.S, and Ph.D. degrees in computer science and system engineering from the University of Miyazaki, Japan, in 2003, 2005, and 2009, respectively. He is currently an Associate Professor with the Faculty of Engineering, University of Miyazaki, Japan. His research interests include computer networks and security. He is a member of IPSJ and IEICE.

Naonobu Okazaki

Naonobu Okazaki received his B.S, M.S., and Ph.D. degrees in electrical and communication engineering from Tohoku University, Japan, in 1986, 1988 and 1992, respectively. He joined the Information Technology Research and Development Center, Mitsubishi Electric Corporation in 1991. He is currently a Professor with the Faculty of Engineering, University of Miyazaki since 2002. His research interests include mobile network and network security. He is a member of IPSJ, IEICE and IEEE.