Image Processing for Web-to-Print Applications

Dmitry Sevostyanov

Keywords: algorithms, automation, HTML, PDF, output

Abstract

This article highlights approaches to building interactive online editors that work in a browser and enable print design creation. Rendering graphics is key to properly displaying products in a browser. Customer's Canvas has shifted many image processing tasks to the front end, which has increased the speed of our editors, boosted the quality of graphics, and improved matching the design in the browser with the printed version. Our approach allows us to process designs on the front end with fewer requests to the server and without sacrificing performance. These innovations were recognized with a 2020 InterTech Technology Award from Printing Industries of America.

Introduction

As with any production process, printing consists of several actions, and the quality of printed products depends on how well each of these actions is implemented. These steps include design creation, prepress, press, and post-press processing. The design must be accurately transferred from the editor to a printable file to ensure that it is printed correctly.

Printers originally solved image-processing tasks with desktop publishing software. The main approaches we use today to transform designs into print-ready files were developed over several decades. Although the emergence and development of online ordering processes for printed products like web-to-print have revolutionized the industry, these helpful tools have led to new challenges.

This paper will examine the main features of image processing in web-to-print systems. We will especially focus on what distinguishes this technology from desktop publishing systems, the likely points of difficulty, and how the Customer's Canvas development team responds with solutions that were judged worthy of a 2020 InterTech Technology Award from Printing Industries of America.

Aurigma; InterTech Award Recipient

Image processing for web-to-print applications: challenges and limitations

Although web-to-print systems and desktop publishing applications perform relatively similar image processing tasks, they are very different from a technical standpoint.

The two most distinct differences are:

1. Distributed image-processing operations.
2. Restrictions imposed by web-based technologies.

Distributed image-processing operations. Desktop publishing systems are usually comprised of the following components: graphics, user data, the design process, and the process of rendering the output file. These components are all in one place – on the computer where the design is created. This is not the case when it comes to web-to-print systems.

Different systems approach this task in various ways, but in most cases, the source graphics, user data, and code for creating the output result are located on the server. The user interface for editing is located in the browser on the client's computer. The resulting file must also be generated on the server, where all the logic of working with personalized designs is located. The distributed nature of these components in space and time imposes serious restrictions on creating a user-friendly editing process.

Restrictions imposed by web technologies. Web browsers have been created to display and edit online content but they are not designed to work with graphic design. The HTML format lacks the full range of functionality that web-to-print systems need. We must employ special methods to circumvent these limitations.

Web-to-print systems were initially conceived as a way to streamline the costs of printing companies for receiving and processing orders. The printing company must consider the seamlessness of the user experience as well as the quality of the final product as it implements these systems.

Since these are frequently two opposing tasks, developers of application software for editing printed products online need to address the pitfalls that do not apply to desktop applications.

These pitfalls can be bundled into several categories:

1. Performance of the online editor.
2. Precise design transfer from the editor to the print file.
3. Implementing imaging features unavailable in standard web technologies.
4. Characteristics of printing technologies.

Performance of the online editor

The online editor's speed is a major determining factor that affects the usability of the online design editing process and, by extension, the chances of successful order fulfillment. In this sense, web-to-print solutions are quite different from online editors, which are geared toward creating designs for later use in digital formats. Since printing is considerably more sensitive to quality and final image resolution, there is a risk of a significant reduction in the editor's performance in the browser. This is particularly relevant to large products like signage.

If templates and design elements were downloaded in their original size to the client's browser, it would put the strain on the customer's end of downloading hundreds of megabytes in some cases. To further complicate matters, when this data is downloaded, there is a high chance that the browser will lose a lot of performance where the editor is running, which will make the editing process slow and unresponsive. This situation has a high chance of frustrating the customer enough to close the page and seek out a variable order method or supplier instead.

To cope with this task and preserve the ability to create hi-res files without compromising usability, a two-tiered editing approach is implemented in web-to-print systems.

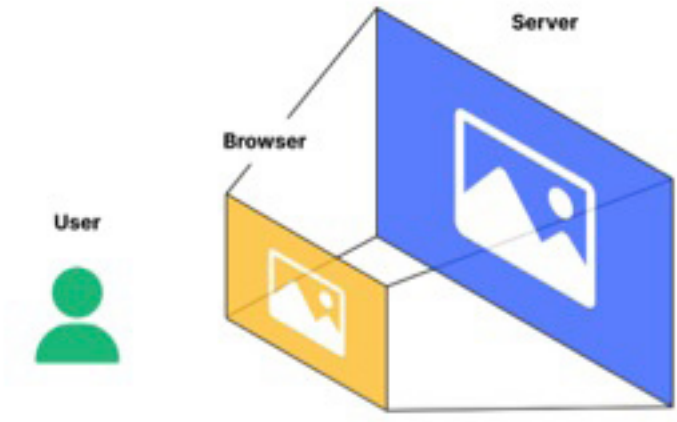

Figure 1: *Front-end and back-end versions of a product design*

The customer works in the browser with a copy of the product, which is optimized for their screen size and provides reasonable traffic consumption, high performance, and responsiveness. This is very similar to the way that desktop software handles editing. To improve performance, users edit a copy of the design, and a full-size design is rendered at the end.

The same principle applies to web-to-print. The big difference is that desktop software processes images that are usually stored in the same location, while in the use case of web-to-print, the full version and its copy might be thousands of miles apart. The full template version in its original size is stored on a server. When the user has made all the necessary manipulations to their design, information about the changes applied to the design is sent to the server where the design is rendered into a ready-to-print file.

For this setup to work, an image-processing engine runs on a server and has the option to process the low-res copy on the client-side. The engine automatically reduces all the graphics before sending it to the client and loading it into their browser. The server then receives the smaller version of the created design and applies all the changes made by the user to the full-sized original file.

To improve an editor's usability, image-processing engines in web-to-print systems can sometimes cope with a task that is not typical of desktop engines. This solution defers the generation and delivery of graphics to the browser to minimize the amount of data transmitted to the client before loading the editor, which will improve the user experience and make the editor faster and more responsive. Loading the editor on the page and loading the design in the editor are separate events and operate asynchronously. This means that when the editor is loaded, the server starts to prepare the design and passes it in parts to the editor, where it is displayed.

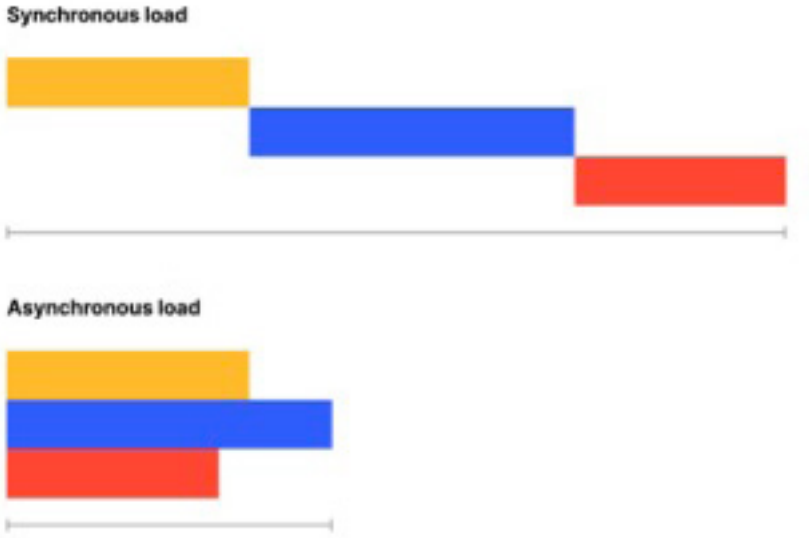

Figure 2: *Synchronous and asynchronous load overall scheme*

Another method for increasing the online editor's productivity is to reduce the number of requests to the server. To implement this approach, it is beneficial to have two processing engines that work with graphics equally. One engine on the server-side can render hi-res files, and another on the client-side can demonstrate to end-users how their design will appear. The result will be the same in the editor and in the hi-res file that is rendered on the server.

In practice, this means porting the engine to the client because the built-in rendering tools, which run in the browser, are not capable of working with graphics and text at the level required for professional design creation.

Precise design transfer from the editor to the print file

Once the end-user has finished editing and approved the final product appearance, it is important to transfer their design to the print-ready file as accurately as possible.

Pixel-perfect rendering and screen coordinates. As we mentioned earlier, the end-user works with a smaller copy of the design in their browser. This presents another problem: the design is created at one scale, but the output file will be rendered at another. This can lead to unsightly artifacts, like a slight shift of the design elements in the final output. For example, there may be a tiny gap between elements only a few pixels wide. It might seem like a minor issue, but this can be enough to shift a word to another line in some cases.

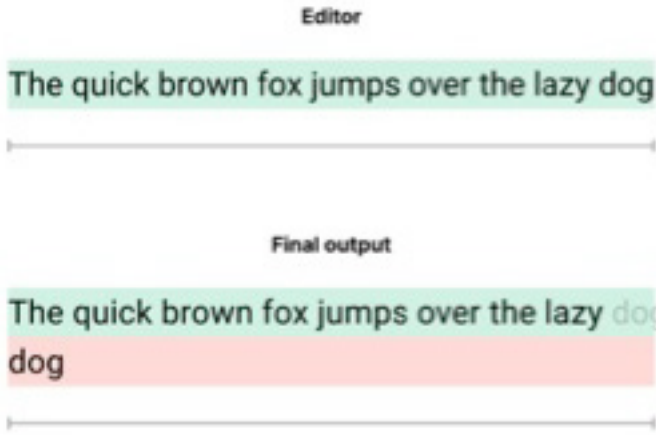

Figure 3: *Example of possible design flaw caused by different rendering engines in a browser and on a server.*

This problem can also occur when vector and raster elements are used in the same design. The image-processing engine should be designed to avoid this type of problem with fractional pixels.

Color management. Web browsers were not designed for printing applications, and they are only capable of supporting the RGB color space. On the contrary, printed designs are stored in combinations of process colors – Cyan, Magenta, Yellow, and Black (CMYK) – which are directly supported by the printing equipment. Consequently, customers must create CMYK designs in an editor that only works in RGB.

A special color management technique called proofing makes it possible to view a print for a specific type of printer on the screen to understand how it will look. This technique can be used to organize the process of creating CMYK designs in the web-to-print editor. In turn, the image-processing engine must be designed to support it.

Implementing imaging features unavailable in standard web technologies

Text engine. Working with text is one of the most complex tasks of a web-to-print editor. These difficulties are related to the limitations of web technologies.

As we learned before, web browsers were created without much regard to the process of editing graphics for print. This limitation has imposed many restrictions on how we can work with fonts. Fonts have a long history in the printing business. Many font families support complex metrics, affect the display of glyphs, and allow one to flexibly configure the font style.

As web browsers developed over time, decisions were made to abandon some font formats and to abolish a large number of metrics. Therefore, it can be complicated to facilitate editing complex calligraphic designs in browsers.

These days, Adobe products are recognized as the de facto standard for the printing industry. They have a much more sophisticated text engine with many specific features that standard HTML text lacks: the wide range of text arrangement types, different alignments, scales, open type, auto-size features, and so on. The text engine should support as many of them as possible.

Web-to-print systems that require support for complex text features use two main approaches:

1. Render the text on the server with an engine that supports all the features for working with fonts.
2. Use code that executes on the user's browser for getting proper text layout.

The first approach operates in most web-to-print systems, but it is not user-friendly. This is because end-users edit the text in a pop-up window, after which the text will be rendered and added to the design as an image. This is often a time-consuming

process, especially for long paragraphs. Customers end up enduring delays as they edit their designs.

The second approach is significantly more user-friendly and provides pixel-perfect rendering during the editing process and in the output design. However, this is more complicated to develop. To implement this approach, text must be drawn on the client-side in the same way that it is executed on the backend.

Standard HTML5 technologies aren't designed for this purpose. There is no glyph support available and the web text-rendering engine does not have enough text settings to edit complex, text-centric designs like brochures. Moreover, each browser has a unique engine that processes the images in a slightly different way. In certain browsers, the typography may vary and other features may not work in the same way.

Image editing. Some types of products and workflows require the ability to edit images before adding them to the design. This can include various effects, such as geometric transformations of images, color management, color reduction, and so on. In the classic approach to this task, the end-user first uploads the image to the server, and then the image is edited, returned to the editor, and finally added to the printed design. Adding another image-processing engine to the client will speed up the process and upload an edited image directly to the web-to-print editor, bypassing the server.

3D preview and AR support. If a customer orders a product with a complex shape, such as packaging, it's difficult for them to mentally visualize what the finished product will look like. The web-to-print software helps the customer out by rendering the final version as a 3D model. Consequently, a full-fledged 3D engine is built into the browser.

Characteristics of printing technologies

Color reduction. Some types of products and printing technologies are incompatible with full-color designs and require reducing the number of colors used to a certain number. In web-to-print systems, the difficulty of the situation is compounded by the end-user working with graphics on the client with a need to see these reduced colors in the browser where the editor is loaded. The image-processing engine should implement this functionality and ideally reduce the colors on the client without accessing the server. The color reduction algorithm is not trivial, especially if it's necessary to emulate an infinite space of colors with a limited palette.

Two main tasks need to be solved here:

- Make a list of a small number of colors (2, 3, 4, or 6) from a large palette based on the design.
- The color reduction algorithm, which should take into account both raster and vector elements.

It's important to keep in mind that printing technologies that require color reduction create additional limitations, for example, they do not support small-scale elements.

Blending modes. Blending modes are among the many limitations of browsers in terms of image processing functionality for print applications. If you have used Adobe software before, then you know that there are a large number of options and settings that determine how semi-transparent images placed on top of one another can be combined into the output result. The browser does not support all these blending modes, so the image-processing engine must pick up the slack and specially emulate the modes to display them on the client and achieve a realistic result in the editor. Otherwise, pixel-perfect rendering is not possible in the web-to-print application.

Post-press processing simulation. Printed goods are physical objects that are sometimes hard for the customer to visualize as they work on a flat design in the editor. Ideally, customers would like to see a realistic preview of how their design looks on a product. This is a standard function of web-to-print systems in most cases. However, post-press processing like foiling or spot varnish can complicate matters. For such products, the image-processing engine must emulate how the design will look if a selected effect is applied to it.

Conclusion

As demand grows in the printing industry for web-to-print solutions, the requirements for tasks solved using image-processing technology will increase as well. The emergence of more complex use cases implies that image processing is improving. Accordingly, online solutions will get closer to the same level of desktop software in terms of print-ready file rendering capabilities.

More technological constraints will need to be bypassed to approach the technological parity of online and desktop systems. Experience has shown there are many ways to circumvent restrictions that often incur large development costs. Thus, the advanced functionality will be mainly available to enterprise customers first. Nevertheless, enterprise companies most frequently make increased demands on the quality of the final product and rendering accuracy.

Advanced print product design functionality is gaining momentum as more software becomes available online. This trend drives more vendors than ever to implement

the latest image processing features to make the design process feasible for online customers. Ongoing innovation in image processing is one of the key factors that ensure a competitive edge for web-to-print software developers.