



A MULTI-LAYERED IMAGE FORMAT FOR THE WEB WITH AN ADAPTIVE BACKGROUND LAYER SELECTION ALGORITHM

Milan Tair,
Aleksandar Mihajlović,
Nikola Savanović

Singidunum University,
32 Danijelova Street, Belgrade, Serbia

Abstract:

In this paper, we present a proposed multi-layered image format for use on the web. The format implements an algorithm for selecting for rendering one of two or more layers adaptively. The selection process depends on the weighted average brightness of the underlying web page background within the image bounds. The proposed image format supports multiple image layers with adjoined thresholds. Depending on these thresholds and the underlying background, a layer will either be shown or hidden. The selection algorithm takes into account the background brightness and each layer's adjoined threshold values.

Key words:

image format, layers, selection algorithm, website.

1. INTRODUCTION

Ever since the early days of the Internet, programmers have been inventing new ways of including multimedia into web pages. As connection speed used to be slow, image formats were designed with limitations to conserve space. This also helped reduce the impact of loading extra content into the page over the network. Some formats introduced colour indexing (Miano, 1999). This provided a way to use a greater number of colours without restricting them to a 8-bit colour scheme. Other compression algorithms are designed on the fast Fourier and discrete cosine transformations (King, 1995). Nevertheless, both of these formats still had major limitations despite providing smaller file sizes. Among these limitations was the lack of the alpha channel. Indexed colour image formats, such as the Graphics Interchange Format (GIF), introduced binary transparency. This feature provided the ability to use limited transparency capabilities in image formats. That was the first step before the introduction of full alpha channel implementations. Subsequent Web media formats, such as the PNG format, introduced four values per pixel. Three values are for the red, green and blue colour components and the fourth is the alpha channel (Powers, 2008). What all these formats have in common is the fact that they are all single layer image formats. Layered image formats exist and are commonly used in print and pre-press (Gatter, 2005). These formats have not seen much use on the Web. This is because of limited browser support (Comparison of web browsers - Image format support, 2016) and inconsistent image layer rendering techniques. This results in

Correspondence:

Milan Tair

e-mail:

milan.tair@gmail.com



unpredictable displaying across different browser engines (Chang, Yen, & Hsu, 2008). Even formats with alpha channel support have the issue of contextual adaptability. Layered image formats are rendered in such a way that all layers are placed one on top of another. The result would be the same as in the case when the image was created within a pre-rendered single layer image, stored with multilevel an alpha channel.

2. PROBLEM DEFINITION

The aim of this paper is to introduce a new image format. This format attempts to resolve a problem of low image content to background contrast. This problem occurs when images with alpha channels are placed on variable brightness backgrounds.

For example, there is an image of a certain company's logo. The logo contains text in a custom type and a certain graphical component. The content of the image is predominantly in a darker colour scheme. This image implements alpha channel transparency. All pixels that do not contain logo text and graphical components are transparent. Such an image is placed over a predominantly bright background. In that case, the content would be easily distinguishable on the page. However, if the web page section on top of which the image was to be placed is predominantly dark, the image contents would be hard to distinguish. The content would be unrecognisable due to low contrast between the background and the content. In such cases, designers would need to create a new logo with predominantly light text and graphical components, which requires more time and professional engagement. In addition, designers should be equipped with an additional document that describes the proper use of different colour versions of the logo. This way, the contrast between the logo and the background can be high enough. Otherwise, the designer may choose to introduce an underlying bright wrapper that would exist within the image itself. This container would surround the main textual and graphical components of the logo. When such an image is placed on a dark background, the surrounding container would be visible. The logo would now remain distinguishable. On the other hand, if such an image were to be placed on a bright background, the container would not increase the visibility of the logo. Instead, it would create a visual obstruction of the underlying background. Also, companies often insist on a minimum clearance margin around the logo and other elements, which can be predefined and added to the file. These are all important parts in creating an automated

file that will ensure the proper use of corporate logo in accordance with all defined guidelines. For companies and organisations, this issue is a key element for ensuring proper brand identity and the immediate representation of the company. A valuable corporate asset must be used consistently in the proper, approved forms to ensure corporate marketplace recognition. Thus, this issue is identified as one of the major issues in the domain of marketing.



Illustration 1. Comparison of rendering a random company logo with opacity via alpha channel on light and dark backgrounds

As can be seen in Illustration 1, the logo is distinguishable on the light background. However, on the dark background, the logo is hard to recognise. Currently available image formats, either single or multi layered, do not provide a solution for this scenario. The current best practice is to create two images and load one or the other depending on the background.

3. PROPOSED SOLUTION

As explained in the previous section, current multi-layered image format implementations do not provide a method of having one or more layers algorithmically selected to be displayed depending on the underlying background. The image format proposed in this paper attempts to solve this. We also describe a simple algorithm for deciding whether a certain layer should be rendered on the page. The algorithm takes a number of arguments for layer selection.

Image format description

The proposed image format is based on the PNG image format. It implements the concept of using an additional value for mapping pixel opacity at a certain point in the image matrix. Also, the format introduces layers with additional meta information and image data. Each layer is described by an optional layer name and two



mandatory brightness threshold values. The algorithm for the layer selection procedure, described in this paper, uses these values. Image data for each layer consists of a complete PNG image with same dimensions as the image. Image format packaging is based on a ZIP archive (Saxena, 2009) with a predefined file structure. The root of the archive contains a JSON (Ecma International, 2011) formatted file manifest.json. The manifest contains plain text description of the package.

```
{
  "format_version": "0.1.0",
  "author_name": "Forename Surname",
  "made_with": "DEMO image editor",
  "use_algorithm": "default-1",
  "dimensions": {
    "width": 165,
    "height": 85
  },
  "properties": {
    "margins": [ 5, 5, 5, 5 ]
  },
  "layers": [
    {
      "min_brightness" : 0.0,
      "max_brightness" : 1.0
    },
    {
      "min_brightness" : 0.0,
      "max_brightness" : 0.3
    }
  ]
}
```

Listing 1. Sample content of the manifest.json file in the archive root

Listing 1 provides the basic information about the image. Available information includes layer thresholds and dimensions of the image. The number of elements in the layers array determines the layer count. In addition, the use of the algorithm property defines which algorithm should be used for layer selection. Introduction of this property reflects the design decision for future implementation of alternative algorithms. Currently, only algorithm 1 is available and described herein. The format version property identifies which renderer engine should handle image rendering.

Alongside the manifest.json file, each layer is stored as an PNG image. Layer images are stored in the root of the archive and are named layer_01.png, layer_02.png, layer_03.png *etc.* There can exist up to 99 layers according to the current version of the proposed format. The enumerated part of the layer filename corresponds to the index in the manifest objects property layers.

Another important element in the manifest is the properties object and its margins value. This value represents an array that can be used to force the rendering engine to allow a clearance margin around the image content. Array values are margins listed in the following order: top, right, bottom and left, *i.e.* clockwise from the top margin, measuring in pixels.

Our solution requires a method of analysing the image clipped from the page background content bounded by our image format destination rectangle bounds. There are many methods of analysing images and processing them in order to determine their properties. In this implementation, we have used an image-processing algorithm that has shown, through experimentation, that it is adequate for the task. We have used a technique of comparing images based on their properties layer by layer and identifying image brightness as well as patterns in the image. Upon identifying patterns, we have clustered them into a weighting matrix used to weight the actual segment of the background in relation to the logo layer stored in the image format.

Layer selection algorithm

In this proposed image format implementation, the drafted layer selection algorithm is relatively simple. It uses the image analysis algorithm mentioned above. First, it finds the average background-brightness-intensity. That value is used to determine which layer should be rendered. This decision depends on the defined image layer thresholds and the underlying background.

The min and max brightness properties stand for the smallest and the largest percentage of the background brightness. If the background brightness is between these values, the layer is rendered. Threshold values are inclusive if there is a layer that has the min. brightness set to 0.0 (0%) and the max. brightness of 1.0 (100%). That particular layer will be shown regardless of the calculated background brightness intensity. As per the manifest example given in Listing 1 for the second layer threshold values, the situation is different. That particular layer will be rendered only if the underlying background brightness intensity is between 0% and 30%. The value 0% is the darkest intensity in the colour scheme. The value 30% is a point, which is at one tenth of the total distance between the darkest and the brightest brightness intensity. More precisely, this layer is set to be displayed only when the image is placed on top of dark backgrounds.

The procedure for calculating the background intensity according to the algorithm 1 implemented for testing this proposal can be described with the following steps:

- ♦ Finding the average opacity of each point in the mask matrix for each alpha channel point of image layers. Equation (1) defines each layer's alpha channel as a matrix of real values of width and height w and h . Each point in the matrix is a value between 0 and 1.



$$L_i = \mathbb{R}^{w \times h}, i = 1 \dots n, n = 2 \dots 99 \quad (1)$$

Equation (2) defines the method of determining values for average opacities of each point in a mask matrix L_α with the same size as layer matrices L_i to L_n .

$$L_\alpha = \frac{\prod_{i=1}^n L_i}{256 * n}, L_\alpha = \mathbb{R}^{w \times h} \quad (2)$$

- Capturing the rendered web page content as a raster image bounded by the size and position where the image is supposed to be displayed. In the current test implementation, this is done using a screen capturing technique based on the third party software (Sofer, 2008).
- Creating a new matrix whose each point represents the greyscale value of each point in the captured raster image of the background. This matrix will be referred to as L_g in the rest of the document.
- Creating a matrix which is the weighted background-worth-matrix calculated by the following equation.

$$L_b = \frac{L_g}{L_\alpha} \quad (3)$$

- After the weighted background worth matrix, its mean is calculated (4) and used as the average background intensity value. We use the mean of this matrix instead of the greyscale background matrix. We value segments under transparent regions more than those under opaque regions of the image layer.

$$a = \overline{L_b} \quad (4)$$

- All layers whose min. brightness thresholds are \leq and max. brightness thresholds are \geq the calculated value are selected to be rendered within the image bounds.

After the selection procedure is done, selected layers are rendered. Layers are rendered one on top of the other within the image element bounds. The first layer is the bottommost.

Active adaptation

The currently proposed image format implementation does not support active background monitoring. It does not include the means of actively calculating the average background-brightness-intensity. Thus, continuous selection of layers to be displayed and hidden on screen is not possible at the time. Future implementations may

provide such a mechanism at greater cost of resources such as memory and processor time.

Predefined image rotation

The proposed image format takes into account the fact that the image can be rotated using CSS3 styling properties. If the rotate property is not applied or is set to 0.0, the rotation procedure is not executed. If the remainder of division of the rotation angle with 360 were to be greater than 0 and less than 360, the rotation mechanism should be executed. This mechanism follows the outlined procedure:

- After the L_α matrix is calculated, its weighted centroids is calculated as explained in (5) and (6). Equation (5) defines the method of determining the x coordinate of the matrix and (6) defines the method of determining the y coordinate of the alpha channel mask matrix.

$$c_x = \frac{\sum_{i=1}^n x_i L_{\alpha_i}}{\sum_{i=1}^n L_{\alpha_i}} \quad (5)$$

$$c_y = \frac{\sum_{i=1}^n y_i L_{\alpha_i}}{\sum_{i=1}^n L_{\alpha_i}} \quad (6)$$

- When the coordinates of the weighted alpha channel centre are found, a new matrix is created. The new matrix width and height are calculated upon applying rotation. The rotation is done around the calculated coordinates. New bounding box coordinates are the minimum and maximum x and y coordinates after applying the original corner point transformation and organising them into new coordinate vector points. A simple coordinate transformation function is given in (7) for the x coordinate of a point. The same method is used for the y coordinate.

$$x_i = x_0 + (x - x_0) * \sin \alpha + (y - y_0) * \cos \alpha \quad (7)$$

In (7) x_0 and y_0 are coordinates of the point around of which the image is being rotated and α is the angle.

- After the new bounds are calculated, the layer selection process can continue from (1). All layer matrices are rotated and fit into the new bounds.

As explained in this section, the main feature of the proposed image format is the ability to define many auto selectable layers. These layers would enhance the visibility of the main content of the image. This way, the image format allows more layer to exist that would be



rendered when the image is placed above the background of a certain brightness. However, the format allows for a different usage derived from the image format design. Such derived usage allows designers to create at least two different images. For example, these may be logo designs. Designers can choose which logo is optimal for different backgrounds. The designer can set the logo, which is optimised for bright backgrounds into a layer set to be rendered when the image is placed on a bright background. Likewise, he or she can set the logo optimised for dark backgrounds into a layer set to be rendered when the image is placed on a dark background. This way, one image package can contain two completely separate images. Each layer contains an optimised logo for the dark and light backgrounds. This method can be used to create one image for more than two situations for different grades of background brightness.

4. USE CASE

In case of the image containing a company logo shown in Illustration 1, at least one solution can be found for resolving the low contrast issue. One method relies on adding a permanently visible surrounding logo holder that would be visible on all backgrounds. This method would be the only applicable in case of PNG images that are the most used image formats on the web, which fully support the alpha-channel (Web Technology Surveys, 2016). In this section, two methods of resolving the aforementioned issue are given.

The discerning container method

The first method uses the proposed image format that has two layers. The topmost layer contains the company logo as seen in Illustration 1. The second layer contains a drawing of a certain shape designed to fit the logo visually. It is designed to be an adequate surrounding container of a light colour combination. It is set to become visible for background brightness of 60% or less. The result of the rendering is shown in illustration 2.



Illustration 2. Comparison of rendering a random company logo with opacity via alpha channel on light and dark backgrounds with the use of the proposed image format with a second adaptively selected layer seen on the right.

As can be seen in Illustration 1, the rendering shown on the right is a result of the second layer being shown in the finally rendered image. This is because the background brightness is less than 60%. The image on the left shows the same logo without the second layer shown. This is because its underlying background, although coloured, is not within the defined threshold range of 0% to 60% for that layer.

The selective content method

This image format also provides the derived use. Initially, this use was not considered. Later on, it was recognised as a potentially good practice. It is characterised by using two or more different image contents for each layer. In addition, layers have threshold values set in such a way so that no more than one layer is shown at one time for one background brightness intensity. For example, this selective content method allows the designer to define two modes of the same logo. Each logo can be optimised for two selected background brightness ranges (Koyani, Bailey, & Nall, 2004). In the example given in this use case, the original company logo has another version adapted for dark backgrounds. The image is packed with two layers. Each layer contains the full logo. Thresholds are set to 0.0 and 0.5999999 for the first layer, which contains the dark background optimised logo and 0.6 and 1.0 for the second layer, which contains the light background optimised logo. The resulting rendering of an image packed in that way is shown in Illustration 3.



Illustration 3. Comparison of rendering a random company logo with opacity via alpha channel on light and dark backgrounds with the use of the proposed image format with two selectively displayed interdependent content layers.

The method used to render the logo shown in Illustration 3, can *i.e.* be used for creating a range of more than two logos. Each logo can be fitted for more than two background brightness intensity ranges. By designing such an image, almost all scenarios can be accounted for. This way, inadequate contrast issues are resolved with just one file loaded into the web page. This is especially useful if the web page design changes themes for each user according to the personalised design (I-Hsien & Hui-Ju, 2009).



Comparing layer images and identifying their peoperties

Each layer image is matched against the background and the appropriate thresholds are identified. The thresholds are selected by identifying dissimilarities between the image and the background. Proper thresholds are identified by measuring the recognisability and similarity of positions of certain peaks.

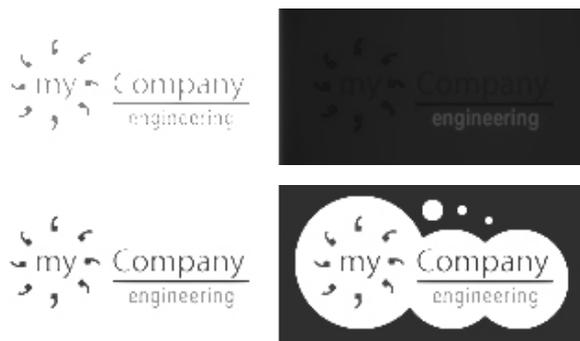


Illustration 4. Comparing peaks β and β' in order to identify patterns within the image resulting from overlaying a layer image over the target background

Parameters that are used for comparison are colours, textures, contours and peaks. Multiple parameters are chosen for comparison because it yields better results. We have compared frequencies for each colour j in the background β and the overlaid image β' based on the coherent pixels for the same colour rounded by the following formula (8).

$$dist(\beta, \beta') = \sum_{j=1}^J \left(\left| \frac{\alpha_j - \alpha'_j}{\alpha_j + \alpha'_{j+1}} \right| + \left| \frac{\beta_j - \beta'_j}{\beta_j + \beta'_{j+1}} \right| \right) \quad (8)$$

In (8) α and β are coherent and incoherent pixel colour histograms, respectively. The presented method was used to calculate the distance between colours.

First, we used a monochromatic version of the resulting image with different brightness levels (grey and white), which were automatically iterated through to create all possible thresholds. In such images, we looked for negligible and regular variations of the distance from the midpoint colour value (grey) and we recorded all repetitions. With these results, we formed a matrix with rows and columns representing all shades of the image and the number of identified peaks. By iterating through this image, we have identified thresholds with the maximum number of smallest coherent peaks. Finding peaks allows for finding the outline of the shape of the rendered image.

After putting the logo over the background, we use the following method:

$$K = \sum_{a=0}^{N-1} \sum_{b=0}^{N-1} (a - b)^2 g(a, b) \quad (9)$$

In (9) a and b are indices of the formed matrix and g is the matrix. Value K is the resulting average colour intensity of the entire matrix.

When we utilize these methods to analyse images for all thresholds in the matrix and their corresponding calculated K values, we found the number of pixels of the matching K value, thus finding all peaks. These peaks are counted for each threshold.

For logos overlaid on over a pure white background, the following peak representations are found:

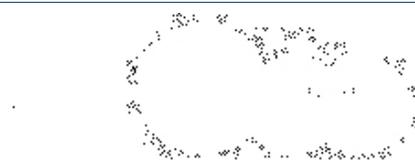


Illustration 5. Comparison of the number of identified peaks at two neighboring threshold values for the logos over a white background

In Illustration 5, we show the peaks for threshold values 45 and 46. The number of peaks at threshold 45 is 1, which is also the minimum useful threshold value, where the image on the right shows a much larger number of peaks found for the threshold value 46.



Illustration 6. Comparison of the number of identified peaks at two neighboring threshold values for logos over a white background

In Illustration 6, we show the peaks for threshold value 48. At this threshold value, the first image on the left shows a smaller number of peaks, while the image on the right has more peaks, but the image has become unusable because the background had covered the entire surrounding of the logo.

When the designer does not specify proper thresholds or wants the application to suggest proper thresholds for the newly designed and packed image, this method is



used. Essentially, we attempt to find the minimal threshold that yields the maximum number of peaks, which are usable. In this way, we identify which of the layers (images) should be used for a predefined set of backgrounds partitioned based on the total number of layers packed in the image of this format. For example, if the image has two layers, we test layers against the white and the black background. If there are three layers, we test them against black, mid-way grey and white.

5. CONCLUSION

As described in this paper, the proposed solution helps solve some issues that occur with image formats used on the web. The key problem solved by the proposed image format is the insufficient content to background contrast distance. It occurs on web sites that change their design based on the user personalisation philosophy. This is especially true in the domain of colour and contrasts of the surrounding context. The low contrast issue is currently solved by creating multiple files. Appropriate files are loaded through CSS depending on the known colour of the background. This solution is flawed in cases when the background is not a known colour. In such cases, the yielding visual impression, derived from its combination with the image is not predictable. Such cases may occur from backgrounds comprised of photographs *etc.* The solution detailed in this paper presented a way of using the proposed image format. By supporting multiple layers with additional meta data comprised of threshold values, the rendering engine determines if a layer should be displayed or not. In addition, three related scenarios were presented that illustrate situations where a random company logo is displayed on bright and dark surfaces. Scenarios with a common PNG image format and the proposed image format are shown. The new image format is presented by using two suggested design techniques to help one image file adapt to both background surfaces better. With both techniques, designers can set additional Meta data that would ensure proper use of the logo in accordance with the predefined rules and guidelines. This would result in a decrease of improper use of corporate logos by third parties, because companies depend on their logo being used in accordance with their defined guidelines for branding as one of the key elements of marketing.

6. FUTURE WORK

The current implementation of the image format uses third party software in order to completely implement the concept. This software requires that the HTML of the web page is copied and remotely converted into an image. In the future, the development of a method for obtaining the underlying background will be researched even though there is no pending implementation of such a functionality in web browser cores. In addition, future work will explore more methods of implementing support for vector image layers. Another important part of our future work will be to identify or develop a better method of identifying optimal layer threshold values.

REFERENCES

- Chang, R.-I., Yen, Y., & Hsu, T.-Y. (2008). An XML-Based Comic Image Compression. *Advances in Multimedia Information Processing 2008*, 5353, pp. 563-572.
- Comparison of web browsers - Image format support.* (2016). Retrieved 01 11, 2016, from en.wikipedia.org/wiki/Comparison_of_web_browsers#Image_format_support
- Ecma International. (2011). *ECMAScript 2015 Language Specification*. Retrieved 01 18, 2016, from ecma-international.org/ecma-262/5.1
- Gatter, M. (2005). *Getting it Right in Print: Digital Prepress for Graphic Designers*. Laurence King Publishing.
- I-Hsien, T., & Hui-Ju, W. (2009). *Web Mining Applications in E-Commerce and E-Services*. Springer.
- King, G. A. (1995). *Understanding and Designing Computer Networks*. Butterworth-Heinemann.
- Koyani, S. J., Bailey, R. W., & Nall, J. R. (2004). *Research-Based Web Design & Usability Guidelines*. Computer Psychology.
- Miano, J. (1999). *Compressed Image File Formats: JPEG, PNG, GIF, XBM, BMP*. Addison-Wesley Professional.
- Powers, S. (2008). *Painting the Web*. O'Reilly Media.
- Saxena, S. (2009). *A First Course In Computers: Based On Windows Xp & Office*. Vikas Publishing House.
- Sofer, N. (2008). *SiteShoter*. Retrieved 01 11, 2015, from nirsoft.net/utills/web_site_screenshot.html
- Web Technology Surveys. (2016). *Usage of image file formats for websites*. Retrieved 01 11, 2016, from w3techs.com/technologies/overview/image_format/all