Detection of perceptually dominant colors in images

Zachary Krejci

Follow this and additional works at: http://www.tdcommons.org/dpubs_series

Recommended Citation
http://www.tdcommons.org/dpubs_series/652

This work is licensed under a Creative Commons Attribution 4.0 License.
This Article is brought to you for free and open access by Technical Disclosure Commons. It has been accepted for inclusion in Defensive Publications Series by an authorized administrator of Technical Disclosure Commons.
Detection of perceptually dominant colors in images

ABSTRACT

Detection of colors in an image is an important task in many contexts. For example, providers of website hosting services can recommend theme colors for a website based on a logo that is featured on the website. Prior color detection techniques have various shortcomings. For example, prior techniques often detect colors that lack chroma that are unsuitable as website theme colors; identify multiple perceptually similar colors; fail to detect a dominant color or detect colors that do not exist in the image. Further, the techniques make detection errors in the presence of image features such as shadows, anti-aliasing gradients, or compression artifacts.

This disclosure describes computationally inexpensive techniques to accurately detect and return an optimal number of perceptually distinct dominant colors in an image. The techniques are robust in the presence of noise and artifacts in the image, e.g., shadows, anti-aliasing, or use of image compression.

KEYWORDS

- color detection
- image colors
- perceptual color space
- DBSCAN
- theme colors
BACKGROUND

Extracting the perceptually dominant colors from an image presents certain technical difficulties, as explained below with examples.

- **Colors comprising even relatively simple images do not cluster easily**

![Fig. 1: (a) An image (b) color distribution of the image in a two-dimensional color space](image)  

Fig. 1: (a) An image (b) color distribution of the image in a two-dimensional color space

Fig. 1(a) illustrates an image, and Fig. 1(b) shows a breakdown of colors in the image in a two-dimensional color space, using principal component analysis. Fig. 1(b) shows the lack of discernible clusters in color space, even for the relatively simple image of Fig. 1(a). The lack of evident clustering manifests as noise for a classifier that operates on the color space. Besides noise, compression artifacts and anti-aliasing can create complex color spaces.

- **Colors that are close to each other in RGB-space are not necessarily perceptually near each other**

The RGB color space is a color space defined by the chromaticities of the red, green, and blue primary colors. A wide variety of colors are produced by adding these three primary colors.
colors in different amounts. Colors can be represented as vectors with components based on the additive amounts of each primary color.

![Diagram illustrating the difference between RGB and perceptual space](image)

*Fig. 2: Illustrating the difference between RGB and perceptual space*

Fig. 2(a) is an image that is composed of three dominant colors (yellow, green, red). Each color has a corresponding shadow (202). The ideal color identification from this image is simply the three visually dominant colors yellow, green, and red, with shadows not considered as distinct colors. However, even though the shadows are perceptually similar to their primary colors, the two are quite far apart in RGB space. The RGB color space is not perceptually linear.

As a result, naive approaches to color identification, e.g., based on bucketing followed by clustering within RGB space, results in detection of colors as illustrated in Fig. 2(b). As seen
in Fig. 2 (b), the identification of colors doesn’t match color perception from the source image and also introduces colors not present in the original image.

- **Anti-aliasing can result in unwanted detected colors**

  In order to blend an object smoothly into background, an anti-aliasing color is introduced that is in between the colors of the object and its background.

![Illustration of anti-aliasing](image)

**Fig. 3: Illustrating artifacts arising from anti-aliasing**

For example, Fig. 3(a) shows an image which mainly comprises a dark blue color. In order to smoothen the edges of the image and blend it into the background, an anti-aliasing color (light blue) is added. Standard color detection techniques that don’t remove anti-aliasing artifacts, identify both dark and light blues, as shown in Fig. 3(b). The presence of light blue in the set of detected colors is unacceptable, as the light blue used for anti-aliasing is visually imperceptible.

- **A linear gradient between two colors can result in a single detected color**
Fig. 4: A linear gradient between two colors can result in one color going undetected

Fig. 4(a) illustrates an example of linear gradient between two colors (yellow and red), such that one color merges into the other. The colors are uniformly distributed in the source image. It is desirable that two (yellow and red) or three (yellow, red and orange) colors appear in the color-detection result. However, a naive clusterer returns just one color, as shown in Fig. 4(b), even though the two colors in Fig 4(a) are perceptually quite far away from each other.

DESCRIPTION

This disclosure presents robust color detection techniques that return visually dominant colors that represent an image. If two colors are perceptually distinct, then both colors are identified. If a color is a shade of another color, then such a color is not identified separately. Furthermore, colors that lack chroma, e.g., shades of grey, are not returned. For example, when
the detected colors from a logo are to be used theme colors (e.g., for a website), such colors are unsuitable since they typically correspond to text or background colors of the logo.

The techniques presented herein are robust to anti-aliasing, gradients, and compression artifacts. The color-clustering techniques described herein are suited to identifying clusters in the presence of noise. Furthermore, the techniques choose an appropriate number of colors without user input. The techniques are computationally inexpensive and can be deployed as server-side or client-side code.

![Diagram](attachment:image.png)

**Fig. 5: Detection of visually dominant colors from an image**

Fig. 5 illustrates robust detection of visually dominant colors of an image, according this disclosure.

**Transform image to perceptual color space**

The image is transformed from an RGB color space to a perceptual color space (502). A perceptual colors space is one where the Euclidean distance between the vectors representing the colors corresponds to the difference in visual perception between the colors. An example of perceptual color space is the CieLuv color space, in which a color is represented by three
coordinates, denoted $L$, $u^*$ and $v^*$. There is a straightforward mapping between RGB space and CieLuv space. The RGB⇔CieLuv map is applied to each pixel of the image in order to render the image in CieLuv space.

**Bucket image points**

Each point in the image is bucketed (504). As even small images amount to large datasets for the purpose of clustering, for efficient execution of a clusterer, each pixel color is partitioned into buckets. The buckets themselves are clustered treating each bucket as a point in feature space.

An example approach to bucket pixel colors is to shave off the last few bits of precision for each of the $L$, $u^*$ and $v^*$ coordinates and map the pixel color to a bucket indexed by a combination of the shaved-off $L$, $u^*$ and $v^*$ coordinates. To illustrate this procedure, let $B_1$, $B_2$, $B_3$, … , $B_M$ be $M$ color buckets. Denote the number of shaved-off bits by a non-zero integer $F$. Then a pixel with color coordinates $(L, u^*, v^*)$ is placed in a bucket $B_N$ with index $N$ given by

$$N = ((L >> F) << 2 * (8 - F)) | ((u^* >> F) << (8 - F)) | (v^* >> F)$$

where

| is the bitwise logical-or operator;

$\ll$ is the bitwise left-shift operator; and

$\gg$ is the bitwise right-shift operator.

Once bucketing is completed, the Euclidean distance between any two adjacent buckets is perceptually linear. This permits perceptually meaningful distance calculations to be made.

**Cluster the set of buckets**

The set of buckets are then clustered (506) such that clusters that are formed represent different colors in the image. Clustering is performed using robust techniques, e.g., DBSCAN.
with modifications and heuristics to appropriately compensate for the presence of gradients, anti-aliasing and compression artifacts.

Each bucket is treated as a point input to a clusterer such as DBSCAN. DBSCAN works on a set of points by visiting each point and checking the density of points surrounding it. If the density is high enough, the point is added to a cluster, and the cluster expands to neighboring points. For each of those points, the density of neighbor points is checked, and given sufficient density, the cluster expands to those points recursively.

The density of points surrounding a bucket is calculated by counting the number of pixels bucketed into that bucket as well as the adjacent buckets. This number determines if a cluster should expand beyond a given bucket or not.

- **Noise detection**: To achieve robustness against noise, a bucket is marked as noise if the bucket and its neighbors do not contain a minimum threshold of points. A bucket marked as noise is not added to any cluster. Anti-aliasing/compression artifacts tend to have a low density of points surrounding them and excluding these low density buckets improves color detection results.

- **Anti-aliasing detection**: Anti-aliasing effects in the image are detected as follows. HSV (hue, saturation, value) is a three-dimensional cylindrical color space that is a transformation of RGB space. HSV space has the useful property that gradients from a color to white/black/grey lie along a plane crossing the color and passing through the center of the cylinder. The plane is defined by holding the hue value constant for all pairs of saturation and value. If a cluster is about to start in a hue plane that is already associated with an existing cluster, then cluster formation is terminated. When a cluster expands to a new point, the values for the point is calculated in HSV space. The value is
hue tracked so that hues are compared before starting new clusters. In this manner, anti-aliasing effects in the image are detected during clustering, thereby forestalling the appearance of anti-aliasing artifacts at the output of the clusterer.

- **Linear gradients**: To compensate for the presence of linear gradients between distinct colors, the clusterer first sorts each bucket by the number of points within each bucket. The clusterer starts each new cluster at the bucket containing the most number of points that is not already a member of any cluster. The cluster expands from that point (bucket). If a cluster expands too far past the initial point, as determined by a threshold distance, the cluster expansion terminates early. In this manner, the clusterer detects more than one color when building clusters, even if the multiple colors are separated by linear gradients.

During clustering, it is possible that certain points are assigned to multiple clusters. This gives a more accurate representation when determining which colors belong to a cluster, and also when scoring clusters.

**Score clusters and select dominant colors**

Each cluster is scored (508) by counting the fraction of number of pixels in a cluster divided by the number of pixels used during the clustering. Pixels that are marked as noise or low chroma are not counted in the denominator. The dominant, e.g., most common, color from each cluster is selected.

**Examples**

The below examples illustrate the accurate detection of colors using the techniques described above.

**Example 1: Source image with shadows**
As seen in Fig. 6, the techniques described herein accurately detect dominant colors and suppress the colors of shadows, even in the presence of shadows in the source image.

**Example 2: Source image with anti-aliasing**

**Fig. 7: Accurate color detection in presence of anti-aliasing: (a) source image (b) detected colors**  
(Compare with Fig. 3)
As seen in Fig. 7, the techniques described herein accurately detect the dominant color (dark blue) and suppress the colors that are not perceived.

Example 3: Source image with linear gradient

Fig. 8: Dominant colors are detected accurately in the presence of linear gradients (Compare with Fig. 4)

Fig. 9: (a) source image (b) detected colors
As seen in Figs. 8 and 9, the techniques described herein accurately detect multiple dominant colors even in the presence of a linear gradient.

Example 4: Source image with black, white, or shades of grey

Fig. 10: (a) source image (b) detected colors

Fig. 11: (a) source image (b) detected color(s)
As seen in Figs. 10 and 11, the techniques described herein report colors from the image, while suppressing colors without chroma, e.g., the black color of the phrase “R G B” (Fig. 10) and the white color of the background (Fig. 10 and Fig. 11).

Example 5: Selection of suitable number of dominant colors

Fig. 12: Example images with corresponding dominant colors as detected

As seen in Fig. 12, an optimal number of perceptually dominant colors are detected in the several examples. The detected colors illustrate that the techniques disclosed herein are robust to gradients, anti-aliasing effects, noise, etc. Further, detection of black, white, and shades of grey is suppressed.

CONCLUSION

This disclosure describes techniques to accurately detect and return an optimal number of perceptually distinct dominant colors in an image. The techniques are robust in the presence of noise and artifacts in the image, e.g., shadows, anti-aliasing, or use of image compression. The techniques are computationally inexpensive and suitable for use as server-side or client-side code.