Clustering-Based Color Transfer and Harmony Adjustment for Image and Video Aesthetic Enhancement

Aarnav Nagariya (2021112017), Brahad Kokad (2021112004)
International Institute of Information Technology, Hyderabad, India

***Abstract*— This paper introduces a framework for enhancing image and video aesthetics through color harmony and style transfer techniques. Using k-means clustering, the system extracts dominant color palettes from style images and transfers them to source visuals via Radial Basis Function (RBF) weights and harmonic shift algorithms, ensuring smooth and coherent color mapping. Brent's optimization identifies optimal parameters for eight harmony types, aligning with perceptual color balance principles.**

The framework supports both style-driven color transfer and harmony-based adjustments, offering efficient, customization solutions for multimedia applications. This research bridges artistic color theory and computational methods, advancing tools for digital design and cinematography.

I. Introduction

The interplay of colors significantly influences the aesthetic appeal and emotional impact of images and videos. Color theory and the principles of color harmony have been a cornerstone of artistic endeavors for centuries, guiding the creation of visually pleasing compositions. In modern digital media, the need for tools that can intelligently apply these principles is more pressing than ever, as creators seek ways to enhance the visual quality of content for diverse applications such as advertising, cinematography, and social media. This paper presents a novel computational framework that integrates color harmony principles with automated color transfer techniques to enhance the aesthetics of both images and videos.

Color harmony refers to the arrangement of colors in a way that is perceived as pleasing or balanced. It is deeply rooted in human visual perception and has been formalized into various harmony types, such as complementary, analogous, and triadic. These types are defined by the geometric relationships between colors in a color space, often represented as angles in the hue-saturation-value (HSV) or similar spaces [1]. Professional artists usually rely on experience and intuition to choose their favorite harmonic colors. The artist can choose a harmonic set from prescribed sets provided in handbooks [2] or by using an interactive application [3]. Once the set is defined, the artist needs to color or recolor his/her product with this set, a task that can be tedious when the image is complex and contains many colors. Some efforts have been made in creating automated tools that let users recolor a photograph [4]. However, all of these rely on the user to make sensible choices to maintain color harmony to some extent. And they are single image tools, which means that recoloring one image based on other is a tedious task.

In parallel, style transfer has emerged as a powerful technique in computer vision for applying the color and texture characteristics of one image to another image. While most style transfer methods focus on replicating texture or artistic patterns. Though, artistic recoloring remains a domain with a lot more potential.

Our work birdges this gap by providing a tool which automatically harmonizes colors in an image with selectable presets. It can also recolor an image (hereafter refered to as 'original image') based on a reference image. Moreover, we can apply these techniques to seamlessly color grade videos too. Thus, analyzing and enhancing color relationships in visual media. Our approach focuses exclusively on the transfer and harmonization of color palettes. This emphasizes perceptual consistency and aesthetic enhancement without altering the underlying content structure of the source media.

The core contributions of our work are twofold. First, we develop a method for color transfer that uses k-means clustering to extract dominant color palettes from the reference image and map them to the original image. This mapping is performed using Radial Basis Function (RBF) weights, ensuring smooth and natural transitions between colors [5]. Second, we integrate a harmonic shift mechanism that aligns colors with predefined harmony types, optimizing the visual balance of the content. Brent's method, a robust optimization technique, is employed to identify the optimal parameters for applying color harmony to any given visual input [6] [7].

For videos, we extend the approach to process each frame individually while maintaining temporal coherence, ensuring that the aesthetic enhancements remain consistent across frames. The system is computationally efficient and modular, allowing for customization based on user preferences or application requirements.

II. Methodology

The workflow comprises three primary stages: color palette extraction, harmonic alignment, and video processing, each described below.

A. Color Palette Extraction Using K-Means Clustering

To perform color transfer effectively, the system first extracts the dominant color palettes from both the original and reference images. This identifies the most impactful colors that influence the overall aesthetics.

- **Input Data:**
  - The original image or video frame(s) whose colors are to be modified.
  - A reference image providing the target color scheme.
- **Clustering Algorithm:** K-means clustering is employed to identify the dominant colors in the source and style images. The images are transformed into HSV or LAB color space to better capture perceptual differences. [8] [9] [10]
  - The clustering process partitions the image into k clusters, where k is a predefined number representing the dominant colors.
  - The centroids of these clusters represent the dominant colors, forming the respective palettes for the source and style images.
- **Color Matching:** The original image palette is mapped to the reference image palette. This mapping ensures that the original image adopts the reference image's color scheme in a perceptually meaningful manner.

B. Color Transfer Using RBF Mapping

Once the palettes are extracted, the next step is to transfer the colors from the style palette to the source image. Radial Basis Functions (RBFs) are used to ensure smooth transitions and natural blending.

- **RBF Weight Calculation:** RBFs create a mapping function between the source and style palettes. The weights are computed using Gaussian functions centered around the style palette colors.

$$f(x) = \sum_{i=1}^{k} w_i \cdot \phi(\|x - c_i\|)$$

  Here, c_i represents the style color centroids, w_i are the weights, and $\phi(\|x - c_i\|)$ is the radial basis function [5].
- **Application to Pixels:** Each pixel in the source image is remapped to a new color using the weighted function. This ensures a seamless and perceptually consistent transfer of the style image's color scheme.

C. Harmonic Shift Adjustment

The notion of color harmony in this work is based on the schemes developed by Matsuda [11] [12], which descend from Itten's notions of harmony [13] [14], widely accepted in applicable fields involving colors. To enhance the aesthetic appeal, the transferred colors are aligned with a predefined color harmony type (e.g., complementary, analogous, triadic). This process involves adjusting the hues while maintaining the perceived luminance and saturation.

- **Harmony Types:** Harmony types are defined geometrically in the hue-saturation plane of the color space [15]. For instance:
  - Complementary: Colors are 180° apart in the hue circle.
  - Analogous: Colors are adjacent within a 30°-60° range.
  - Triadic: Colors form a triangle with 120° separation.
- **Hue Adjustment:** A harmonic shift is applied to the hues of the mapped colors to align them with the chosen harmony type. The adjustment is parameterized as a function of angular shifts in the hue circle.
- **Optimization Using Brent's Method:** Brent's optimization algorithm is employed to minimize the perceptual error between the original and adjusted hues while maximizing adherence to the harmonic constraints. This ensures the harmonic alignment is both aesthetically pleasing and perceptually plausible.

D. Temporal Coherence for Video Processing

For videos, maintaining temporal coherence is essential to avoid flickering or abrupt changes between frames.

- **Frame-by-Frame Processing:** Each frame of the video is treated as an individual image during the color transfer and harmonic alignment stages.
- **Temporal Smoothing:** To ensure smooth transitions, the system applies temporal smoothing to the mapped color parameters. This is achieved by:
  - Calculating inter-frame differences in color values.
  - Applying a weighted moving average filter to smooth out abrupt changes.
- **Optimization for Coherence:** A cost function is defined to penalize large deviations in color mappings between consecutive frames. This cost is minimized using gradient-based methods, ensuring consistency across the video.

E. Implementation and Modular Design

The entire framework is implemented in a modular manner to allow flexibility and extensibility. Key components include:

- **Preprocessing Module:** Converts input images and videos to the chosen color space.
- **Clustering Module:** Performs k-means clustering for palette extraction.
- **RBF Mapping Module:** Handles the computation and application of radial basis functions.
- **Harmonic Adjustment Module:** Implements harmonic shifts and optimization.
- **Postprocessing Module:** Converts the results back to the original color space and formats.

F. Evaluation Metrics

To validate the effectiveness of the methodology, both qualitative and quantitative metrics are employed:

- **Qualitative Metrics:** Visual inspection to assess aesthetic improvements.
- **Quantitative Metrics:**
  - **Perceptual Distance:** Measures color consistency using ΔE in the LAB color space.
  - **Temporal Coherence:** Evaluates smoothness of transitions using frame-to-frame color differences.

This methodology provides a robust and adaptable framework for automated color harmony and transfer, with applications spanning image editing, video post-production, and creative content generation.

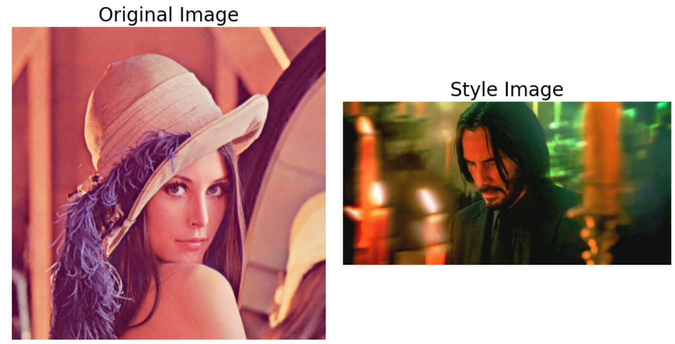

Fig. 1. Original image and style image used for color transfer.

III. Results

This section presents the results of the proposed color harmony and transfer methodology, evaluated using multiple examples and different parameters. The experiments include color palette transfer between a source and style image, variation in the number of clusters k, and application of various harmony types. Visual and quantitative analyses are provided for the outputs.

A. Color Transfer: Source and Style Image

The first experiment demonstrates the transfer of the color palette from a style image to a source image. Figure 1 shows the input source image and the corresponding style image. The objective is to map the style palette onto the source while preserving visual coherence and natural transitions.

B. Effect of Varying k: Number of Clusters

To evaluate the impact of the number of clusters k on the output, we varied k across values 3, 5, 7, and 11. Figure 2 illustrates the outputs for each value of k. As k increases, finer details of the palette are captured, resulting in more precise and varied color transfers. However, excessive clustering (e.g., $k = 11$) can lead to overly complex mappings that may reduce perceptual simplicity.

Fig. 2. Effect of varying the number of clusters k on color transfer. Results are shown for $k = 3$, $k = 5$, $k = 7$, and $k = 11$.

C. Additional Examples of Color Transfer

To validate the generalizability of the proposed methodology, an additional pair of source and style images was processed. Figure 3 displays the second set of source and style images, while Figure 4 shows the corresponding outputs for different values of k. Similar trends in the quality and detail of the transfer are observed, highlighting the robustness of the approach.

Fig. 3. Another set of original and style images for color transfer.

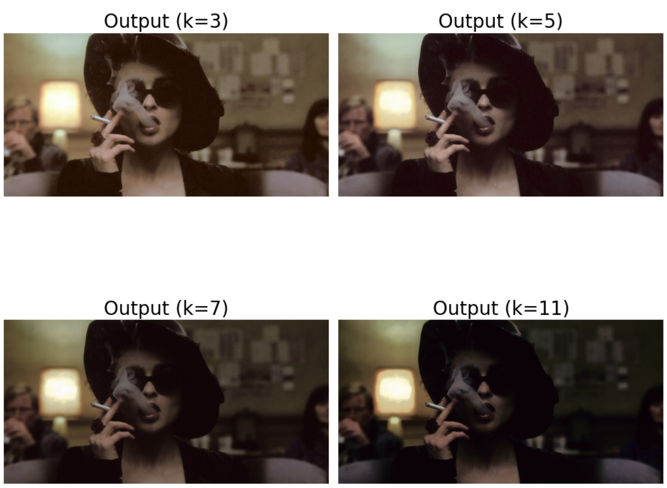

Fig. 4. Effect of varying the number of clusters k on color transfer for the second example, for $k = 3$, $k = 5$, $k = 7$, and $k = 11$.

D. Application of Harmony Types

The effectiveness of the harmonic shift adjustment is demonstrated through the application of various harmony types. Figure 6 presents an image processed with different harmony types, including *i*, *L*, *mirror-L*, *V*, *I*, *Y*, *X*, and *T*. Each harmony type aligns the hues according to predefined geometric configurations in the color wheel, resulting in distinct aesthetic outcomes.

The quantitative values for each harmony type are summarized below:

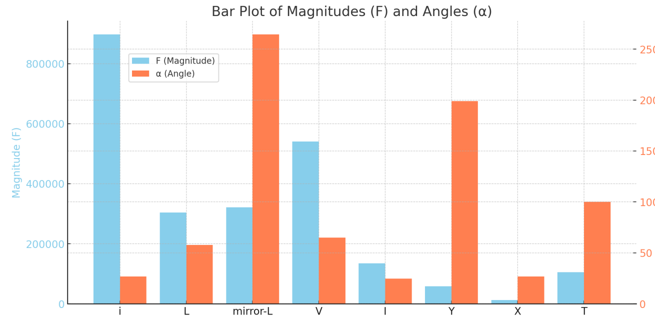

Fig. 5. F and α values for various harmony types.

- **i type:** $F = 897,458.77$, $\alpha = 26.95°$
- **L type:** $F = 303,996.44$, $\alpha = 57.66°$
- **mirror-L type:** $F = 321,240.48$, $\alpha = 264.38°$
- **V type:** $F = 540,343.23$, $\alpha = 64.92°$
- **I type:** $F = 134,615.90$, $\alpha = 24.84°$
- **Y type:** $F = 58,875.53$, $\alpha = 198.98°$
- **X type:** $F = 12,929.27$, $\alpha = 26.72°$
- **T type:** $F = 105,290.49$, $\alpha = 100.08°$

The results reveal that the *X type* yields the lowest objective function value, indicating superior aesthetic optimization under the given criteria. In contrast, the *i type* and *V type* result in higher function values, suggesting their applicability for different aesthetic preferences.

E. Observations

- The proposed methodology successfully maps style palettes onto source images while preserving perceptual coherence.
- Increasing k leads to more detailed color transfer but may reduce perceptual simplicity beyond a certain threshold.
- The harmonic adjustment effectively aligns hues to achieve specific aesthetic effects, with distinct results for each harmony type.

Overall, the results validate the effectiveness and flexibility of the proposed framework in handling diverse input scenarios and aesthetic requirements.

IV. Conclusion and Discussion

This research presents a novel approach to color transfer and harmony adjustment that combines clustering-based palette mapping with harmonic shifts in the color space. The results demonstrate the framework's capability to generate visually appealing and perceptually coherent outputs, with flexible parameters for tuning the desired aesthetic outcomes. Below, we

Fig. 6. Application of different harmony types to an image.

summarize the key findings and insights, discuss limitations, and suggest avenues for future research.

A. Summary of Findings

The experiments reveal several important observations about the proposed methodology:

- **Effective Palette Transfer:** The method successfully maps the color palette of a style image onto a source image while preserving key visual elements. This is achieved through a clustering-based transfer that balances global consistency and local detail.
- **Impact of k Clusters:** Varying the number of clusters, k, allows control over the level of granularity in color transfer. Lower values of k result in broader palette applications with less detail, while higher values increase detail at the expense of potential overcomplexity. The method provides a practical way to balance these trade-offs.
- **Harmonic Adjustments:** Applying different harmony types demonstrates the framework's ability to shift hues according to aesthetic principles, resulting in outputs tailored to specific artistic preferences. Quantitative measures of harmony provide insights into the effectiveness of these adjustments, with the *X type* producing the most optimized results in terms of the objective function.
- **Generalizability:** The framework performs consistently across diverse examples, illustrating its robustness and adaptability to different input styles and source images.

B. Discussion of Key Contributions

The primary contributions of this research include:

1) **Integration of Clustering and Harmony:** By combining clustering for palette extraction with harmony adjustments, the method bridges two complementary approaches to color transfer. This integration enables simultaneous optimization of both aesthetic coherence and color distribution.
2) **Flexibility and Control:** The ability to vary parameters such as k and select different harmony types gives users fine-grained control over the output. This flexibility makes the method suitable for applications ranging from artistic design to automated image enhancement.
3) **Quantitative Evaluation:** The use of objective function values (F) and harmony angles (α) provides a quantitative basis for evaluating the aesthetic quality of outputs. This adds rigor to the traditionally subjective domain of color harmony.

C. Limitations

Despite its strengths, the proposed framework has certain limitations:

- **Computational Complexity:** The clustering-based palette extraction and harmony adjustments can be computationally intensive, especially for high-resolution images or videos. Optimizing the implementation for speed remains an area for improvement.
- **Subjectivity of Aesthetic Preferences:** While harmony types and their objective measures provide a structured approach, aesthetic preferences vary widely among users. Additional personalization options, such as user-defined harmony rules, could enhance applicability.
- **Edge Cases in Color Transfer:** In some cases, the transfer of highly contrasting palettes may lead to unnatural transitions or artifacts. Incorporating perceptual color models could further improve the quality of outputs.

D. Future Directions

Building on the findings of this research, several avenues for future exploration are proposed:

- **Real-Time Applications:** Developing real-time implementations of the framework for video or interactive design tools could expand its practical use cases. Techniques such as GPU acceleration or lightweight clustering algorithms may facilitate this.
- **Improved Perceptual Models:** Incorporating advanced perceptual models, such as those based on human vision or deep learning, could enhance the naturalness and quality of color transfers.
- **Multi-Style Blending:** Extending the framework to support the blending of multiple style palettes could enable more complex and creative outputs, such as gradient transitions between styles.
- **User Customization:** Enhancing user control through interactive tools for parameter adjustment, harmony selection, or manual overrides could make the framework more versatile for diverse applications.
- **Cross-Domain Applications:** Exploring the application of the methodology in domains beyond image processing, such as 3D modeling, virtual reality, or generative art, could reveal new possibilities for creative expression.

E. Concluding Remarks

In conclusion, this research introduces a flexible and effective framework for color transfer and harmony adjustment, combining clustering-based palette extraction with principled aesthetic shifts. By bridging the gap between traditional color theory and computational aesthetics, this work offers a versatile toolset for creators. The method delivers visually pleasing results. Applications include enhancing the emotional tone of visual narratives, creating harmonized themes for branding, and providing automated solutions for post-production editing in the film industry.

By addressing the identified limitations and pursuing future directions, the proposed framework can evolve into a comprehensive tool for creative professionals and automated image enhancement systems alike.

Through this research, we aim to contribute to the evolving field of computational creativity and establish a deeper understanding of how algorithmic methods can amplify human artistic expression.

References

[1] L. Holtzschue, *Understanding color : an introduction for designers*. New York : Wiley, 2006. [Online]. Available: https://archive.org/search.php?query=external-identifier%3A%22urn%3Aoclc%3Arecord%3A1036963024%22

[2] J. Krause, *Color index : [over 1,100 color combinations, CMYK and RGB formulas, for print and Web media*, 2002. [Online]. Available: https://archive.org/details/colorindexover110000krau

[3] B. J. Meier, "Ace: A color expert system for user interface design," *Proceedings of the 1st Annual ACM SIGGRAPH Symposium on User Interface Software,*. [Online]. Available: https://cs.brown.edu/~bjm/p117-meier.pdf

[4] D. Cohen-Or, O. Sorkine, R. Gal, T. Leyvand, and Y.-Q. Xu, "Color harmonization," in *ACM SIGGRAPH 2006 Papers*, ser. SIGGRAPH '06. New York, NY, USA: Association for Computing Machinery, 2006, p. 624–630. [Online]. Available: https://doi.org/10.1145/1179352.1141933

[5] M. D. Buhmann, "Radial basis functions," *Acta Numerica*, vol. 9, p. 1–38, 2000. [Online]. Available: https://www.cambridge.org/core/journals/acta-numerica/article/abs/radial-basis-functions/3FD3A8BBC9B020FA349305142D0EB367

[6] R. P. Brent, "An algorithm with guaranteed convergence for finding the zero of a function," 1971. [Online]. Available: https://maths-people.anu.edu.au/~brent/pd/rpb005.pdf

[7] I. Fortran, W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, "Numerical recipes," *Cambridge, UK, Cambridge University Press*, 01 1992.

[8] D. Bora, A. Gupta, and F. Khan, "Comparing the performance of l*a*b* and hsv color spaces with respect to color image segmentation," 06 2015. [Online]. Available: https://www.researchgate.net/publication/277722637_Comparing_the_Performance_of_LAB_and_HSV_Color_Spaces_with_Respect_to_Color_Image_Segmentation

[9] R. F. M. . M. S. B. Trevor D. Canham, I. Scott MacKenzie, "The effect of perceptual optimization on color space navigability," *Proceedings of Graphics Interface 2019 - GI 2023, Article No. 14, pp. 1-10.*, 2019. [Online]. Available: https://www.yorku.ca/mack/gi2023.html

[10] E. Chavolla, D. Zaldivar, E. Cuevas, and M. Cisneros, *Color Spaces Advantages and Disadvantages in Image Color Clustering Segmentation*, 01 2018, pp. 3–22. [Online]. Available: https://www.researchgate.net/publication/320410000_Color_Spaces_Advantages_and_Disadvantages_in_Image_Color_Clustering_Segmentation

[11] Matsuda, *Color Design (in Japanese)*, 1995.

[12] M. Tokumaru, N. Muranaka, and S. Imanishi, "Color design support system considering color harmony," in *2002 IEEE World Congress on Computational Intelligence. 2002 IEEE International Conference on Fuzzy Systems. FUZZ-IEEE'02. Proceedings (Cat. No.02CH37291)*, vol. 1, 2002, pp. 378–383 vol.1.

[13] J. Itten, *The Art of Color*, 1960. [Online]. Available: https://www.irenebrination.com/files/johannes-ittens_theartofcolor.pdf

[14] D. Briggs and S. Westland, *Itten, Johannes*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 1–3. [Online]. Available: https://doi.org/10.1007/978-3-642-27851-8_340-1

[15] P. Moon and D. E. Spencer, "Geometric formulation of classical color harmony∗," *J. Opt. Soc. Am.*, vol. 34, no. 1, pp. 46–59, Jan 1944. [Online]. Available: https://opg.optica.org/abstract.cfm?URI=josa-34-1-46

V. Appendix

This section will walk you through some explanation of our code and implementation, and is aimed to give a better understanding of the flow.

A. Palettes using K-means Clustering

The provided code implements a version of the K-Means clustering algorithm applied to color quantization. The goal of this approach is to reduce the number of unique colors in an image while preserving the most representative colors. Below is an explanation of each part of the code:

1) Imports: The code uses the following libraries:

- `PIL (Python Imaging Library)`: Used for image manipulation. The `Image` and `ImageCms` modules are imported, though `ImageCms` is not directly used in the code.
- `itertools`: Used to generate combinations of pixel bins.
- `numpy`: Handles numerical operations, particularly for arrays and vectors.
- `random`: Provides random sampling, used to initialize cluster centroids.
- `math`: Contains mathematical functions like exponential and power, used in the attenuation function.
- `util`: Imports two functions, `rgb2lab` and `distance`, which likely convert RGB values to LAB color space and calculate the distance between two colors, respectively.

2) Function `sample_bins`: The `sample_bins` function divides the color space into discrete bins and computes the average color within each bin. It also tracks how many pixels fall into each bin.

- **Input:**
  - `img_pixel_cnt`: A dictionary where keys are RGB pixel values, and values are the count of pixels of that color.
  - `bin_cnt` (default: 16): The number of bins to divide the 256 color levels (for each channel) into.
- **Steps:**
  - The color space is divided into `bin_cnt` bins for each color channel (Red, Green, Blue). The bin range is computed (e.g., for 16 bins, each bin represents a range of 16 color levels).
  - A dictionary `tmp` is created to store the summed color values and pixel counts for each bin.
  - For each pixel, it is assigned to a bin based on its color value, and the summed value and count are updated.
  - The final result (`res`) is a dictionary where the key is the average color of each bin, and the value is the total count of pixels in that bin.

3) Function `init_means`: The `init_means` function initializes the K-means cluster centroids (the ”means”) based on the pixel bins created in the `sample_bins` function.

- **Input:**
  - `bins`: The dictionary of binned colors with their pixel counts.
  - `k` (default: 5): The number of clusters (centroids) to initialize.
- **Steps:**
  - The bins are sorted in descending order based on pixel count (most frequent colors first).
  - The `attenuation` function adjusts the influence of a color bin on the initialization of subsequent centroids, ensuring centroids are spread out across the color space.
  - The function iterates up to `k` times, selecting the most frequent color from the remaining bins for each centroid.
  - The bins are adjusted based on their distance from the most recently selected centroid, using the attenuation function to reduce the influence of nearby colors.

4) Function `k_means`: The `k_means` function implements the K-means clustering algorithm to quantize the image's color palette.

- **Input:**
  - `bins`: The binned colors with pixel counts.
  - `k` (default: 5): The number of clusters (centroids).
  - `init_mean` (default: True): Whether to initialize centroids using the `init_means` function.
  - `max_iter` (default: 1000): The maximum number of iterations for the K-means algorithm.
  - `black` (default: True): A flag to include black as one of the centroids.
- **Steps:**
  - The algorithm starts by initializing `k` centroids using the `init_means` function

or random sampling, depending on the `init_mean` flag.

- If `black` is True, the color $[0, 128, 128]$ is added as a fixed centroid (arbitrarily chosen to possibly represent black or a neutral color).
- For each iteration:
  * Each color bin is assigned to the nearest centroid based on the distance (likely Euclidean distance, computed using the `distance` function).
  * New centroids are calculated as the average of the colors assigned to each cluster.
  * The algorithm terminates if the centroids do not change between iterations.
- Finally, the centroids are sorted by their pixel count, and the function returns the sorted centroids and their corresponding pixel counts.

5) Distance Metric: Although the `distance` function is not provided in the code, it likely calculates the color difference between two points in color space. It may use the LAB color space for the calculation, as the `rgb2lab` function is imported, which suggests that the distance computation aligns with human perception of color differences.

B. LAB to XYZ Conversion

1) 1. Why is the function $f(n)$ defined this way?: The function $f(n)$ is a transformation used to map nonlinear LAB components to linear XYZ components.

a) Definition::

- LAB is designed to model human color perception, where the relationship between luminance and perceived brightness is nonlinear.
- In LAB to XYZ conversion:
  - A **cubic relationship** is used for larger values of n (i.e., $n > \frac{6}{29}$).
  - A **linear approximation** is used for smaller values of n (i.e., $n \leq \frac{6}{29}$) to avoid discontinuities in the curve.

b) Explanation of thresholds and coefficients::

- **Threshold $\frac{6}{29}$:**
  - This value comes from the cube root approximation: $\frac{6}{29} \approx 0.2069$.
  - Below this threshold, the curve becomes nonlinear, and the approximation ensures a smooth transition.
- **Linear approximation coefficients:**
  - For $n \leq \frac{6}{29}$:

$$f(n) = 3 \times \left(\frac{6}{29}\right)^2 \times \left(n - \frac{4}{29}\right)$$

  - Since $\frac{6}{29} \approx 0.2069$, we have $\left(\frac{6}{29}\right)^2 \approx 0.0428$.
  - The factor $\frac{4}{29} \approx 0.1379$ ensures the linear part connects smoothly to the cubic part at $n = \frac{6}{29}$.

2) 2. Why are the constants 95.047, 100.000, and 108.883 used?: These constants correspond to the **reference white point** in the CIE standard. They define the scaling factors for the XYZ color space.

a) Explanation::

- The CIE XYZ color space is based on a standardized *reference white*, which represents the brightest white perceivable under standard lighting.
- The reference white values (X, Y, Z) depend on the specific illuminant and observer conditions. For the commonly used **D65 illuminant** (daylight conditions), the reference white is:

$$X_{\text{white}} = 95.047,\ Y_{\text{white}} = 100.000,\ Z_{\text{white}} = 108.883$$

- These values ensure that the conversion from LAB to XYZ aligns with the perceived brightness of the reference white.

3) 3. Explanation of the equations:

a) Equation for X::

$$X = 95.047 \times f\left(\frac{L+16}{116} + \frac{a}{500}\right)$$

- **$\frac{L+16}{116}$:**
  - Converts the LAB lightness L (0–100) to a normalized value representing Y in XYZ.
  - $\frac{L+16}{116}$ scales the lightness range for further calculations.
- **$+\frac{a}{500}$:**
  - The a-channel represents the green-red color axis in LAB.
  - $\frac{a}{500}$ adjusts X in proportion to how far the color is shifted on the green-red axis.
- **95.047:**
  - Scales the normalized X-component to align with the reference white $X_{\text{white}} = 95.047$.

b) Equation for Y::

$$Y = 100.000 \times f\left(\frac{L+16}{116}\right)$$

- **$\frac{L+16}{116}$:**

- Directly converts LAB lightness L into the normalized Y-component (representing luminance in XYZ).

- **100.000:**
  - Scales the normalized Y-component to align with the reference white $Y_{\text{white}} = 100.000$.

c) Equation for Z::

$$Z = 108.883 \times f\left(\frac{L+16}{116} - \frac{b}{200}\right)$$

- $\frac{L+16}{116}$**:**
  - Normalized lightness for XYZ conversion.
- $-\frac{b}{200}$**:**
  - The b-channel represents the blue-yellow axis in LAB.
  - $\frac{b}{200}$ adjusts Z in proportion to the blue-yellow shift.
- **108.883:**
  - Scales the normalized Z-component to align with the reference white $Z_{\text{white}} = 108.883$.

C. Brent's Method for Hamronization

The following code implements the harmonization of a color palette using Brent's method, a numerical optimization technique. The method adjusts the hues of colors in an image to match a set of predefined target colors while maintaining the overall distribution of the palette's color values.

1) Setup: We start by defining some basic parameters and constants:

- `type_m`: List of palette types (e.g., i-type, L-type, etc.)
- `Tm`: Matrix defining the target color values for each type
- `phi`: Constant for calculating distances

$$Tm = \begin{bmatrix} 0 & 9 & 0 & 9 \\ 0 & 39.6 & 90 & 9 \\ 90 & 39.6 & 0 & 9 \\ 0 & 46.8 & 0 & 46.8 \\ 0 & 9 & 180 & 9 \\ 0 & 9 & 180 & 46.8 \\ 0 & 46.8 & 180 & 46.8 \\ 0 & 90 & 0 & 90 \end{bmatrix}$$

2) Helper Functions: The code includes several helper functions for manipulating and processing color values:

- `shift_hue`: Shifts the hue of an image by a given degree.
- `G`: Gaussian function to weight hue adjustments.
- `arc_dist`, `arc_dist_180`: Functions to calculate the arc distance between two angles.
- `fit_hue`, `real_hue`: Functions to fit and normalize hue values.
- `hue_border_dist`: Computes the distance to the closest target color border.
- `direction`: Determines the direction of color adjustment based on the target type.
- `determine_F`: Calculates the objective function for Brent's method.

3) Brent's Method: Brent's method is an optimization algorithm used to find the minimum of a scalar function. In this case, we use it to find the optimal angle shift for the hues in the palette. The function `brent` implements the following steps:

- Start with three initial points: $a_0 = 0$, $a_1 = 120$, and $a_2 = 240$.
- Evaluate the function `determine_F` at these points.
- Narrow the search interval by comparing the function values and adjusting the points.
- Continue iterating until the difference between the points is less than a threshold.

$$F(a_0, a_1, a_2) = \min\{\text{determine\_F}(a_0), \text{determine\_F}(a_1), \text{determine\_F}(a_2)\}$$

4) Harmonic Shift: The function `harm_shift` applies a harmonic shift to the hues based on the results from Brent's method. It adjusts each color's hue according to the calculated optimal shift, ensuring that the colors move towards their target regions in a smooth manner, while minimizing abrupt changes.

$$H_{\text{new}} = (C_1 - \frac{r_{\text{th}}(w_1)}{2} \cdot (1 - G(\sigma_1, \Delta H)))\%180$$

5) Auto Palette Function: The function `auto_palette` processes the input palette, applies Brent's method to optimize the hue shifts, and then applies the harmonic shift. Finally, it converts the modified hues back to RGB color space.

$$\text{Palette}_{\text{new}} = \text{harm\_shift}(\text{Palette}, M, \alpha, \sigma_w)$$

The final palette is returned in RGB format after being processed in HSV space.