KALMUS: tools for color analysis of films

Yida Chen[1]**, Eric Faden**[1]**, and Nathan C. Ryan**[1]

1 Bucknell University

Summary

KALMUS is a Python package for the computational analysis of colors in films. It provides quantitative tools to study and compare the use of film color. This package serves two purposes: (1) various ways to measure, calculate and compare a film's colors and (2) various ways to visualize a film's color. We have named the software KALMUS in homage to Natalie Kalmus (1882 - 1965), a Technicolor Director who oversaw the color palettes of nearly 300 Hollywood feature films.

Statement of Need

"Colors are elusive"

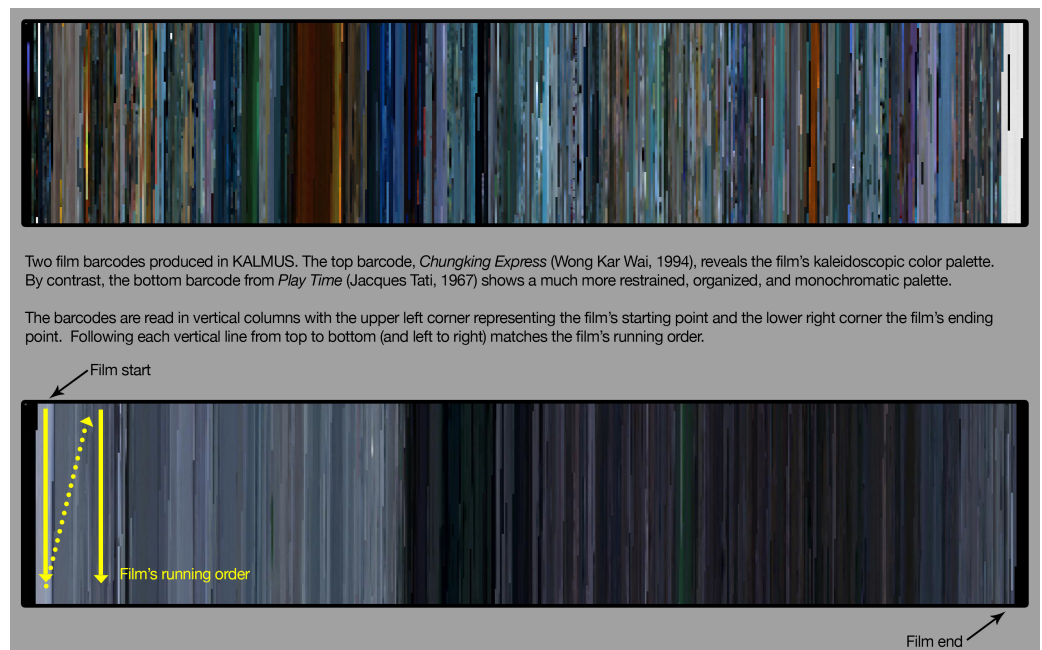Barbara Flueckiger (2017)

"Color is in so many ways uncontainable."

Josh Yumibe (2012)

The epigraphs above (each from the introduction of recent research on cinematic color) all acknowledge the challenges of studying cinematic color aesthetics. Filmmakers, cinematographers, production designers, and colorists spend significant money, time, and labor deciding a film's color palette before, during, and after production. Given this obsessive attention, why is color so "elusive" to study and analyze?

Three reasons emerge: (1) film color remains subtle and almost subliminal by design; (2) we read color subjectively and are often influenced by cultural or even historic circumstances; (3) films are kinetic with the image (and colors) constantly moving. Color might be relatively easy to observe in a single frame, but tracking the color palette across an entire film proves daunting. Moreover, comparing one film's palette to another proves especially challenging due to the accumulation of data needed to make a meaningful comparison. See Stutz (2016) for an overview of both the history and the state-of-the-art of the study of colors in film.

Recently, several digital tools have emerged that analyze a film's color. The ability to capture film stills from DVD or streaming sources and analyze the frame's palette in off-the-shelf tools like Adobe Color or Adobe Photoshop provide insights into individual frames. Additionally, many open-source projects provide implementations of state-of-the-art computer vision algorithms applied to moving images. Some, such as the Distance Viewing Toolkit (Arnold & Tilton, 2020), are very general and handle several aspects of a moving image at once (sound, color, camera angle, etc.). Others, such as VIAN (Halter et al., 2019), allow for a "closer," more interactive distant viewing.

KALMUS is built around a visualization of films known as movie barcodes; it produces an overall color palette "snapshot" of a film. Artists like Jeffrey Moser or programmers such as Charlie Clark, generate movie barcodes by reducing each film frame to a single color[1] and then stitching these colors into a mosaic. See Figure 1 for examples of movie barcodes generated by KALMUS. Existing movie barcode software is inadequate for studying films in a quantitative way. While such software provides a fascinating visual "overview" of a film's color palette, it, unlike KALMUS, focuses on visualization rather than providing meaningful analytical data. KALMUS generates barcodes but also generates a wealth of data related to color and the kinds of barcode and data that can be generated are fairly customizable so that users can carry out their own analyses.


Two film barcodes produced in KALMUS. The top barcode, *Chungking Express* (Wong Kar Wai, 1994), reveals the film's kaleidoscopic color palette. By contrast, the bottom barcode from *Play Time* (Jacques Tati, 1967) shows a much more restrained, organized, and monochromatic palette.

The barcodes are read in vertical columns with the upper left corner representing the film's starting point and the lower right corner the film's ending point. Following each vertical line from top to bottom (and left to right) matches the film's running order.

Figure 1: Movie barcodes generated by KALMUS.

KALMUS allows for the analysis of a film's color by:

1. Providing an interface that takes in either a video file or a JSON file (a sample JSON file can be found here).

2. Allowing for the computation of a frame's color in a number of ways (dominant, median, etc).

3. Allowing for the computation of a film's color in a number of ways.

4. Providing implementations of ways to compare colors of two films.

5. Providing implementations of ways to visualize the color of a film.

6. Allowing the user to download color data as a CSV or JSON file. See Figure 2 for an explanation of KALMUS's user interface.

---

[1]In OpenCV (Itseez, 2015), for example, if a frame is decimated to a single pixel, its color is determined by area interpolation, a weighted average of the RGB-values of the pixels.
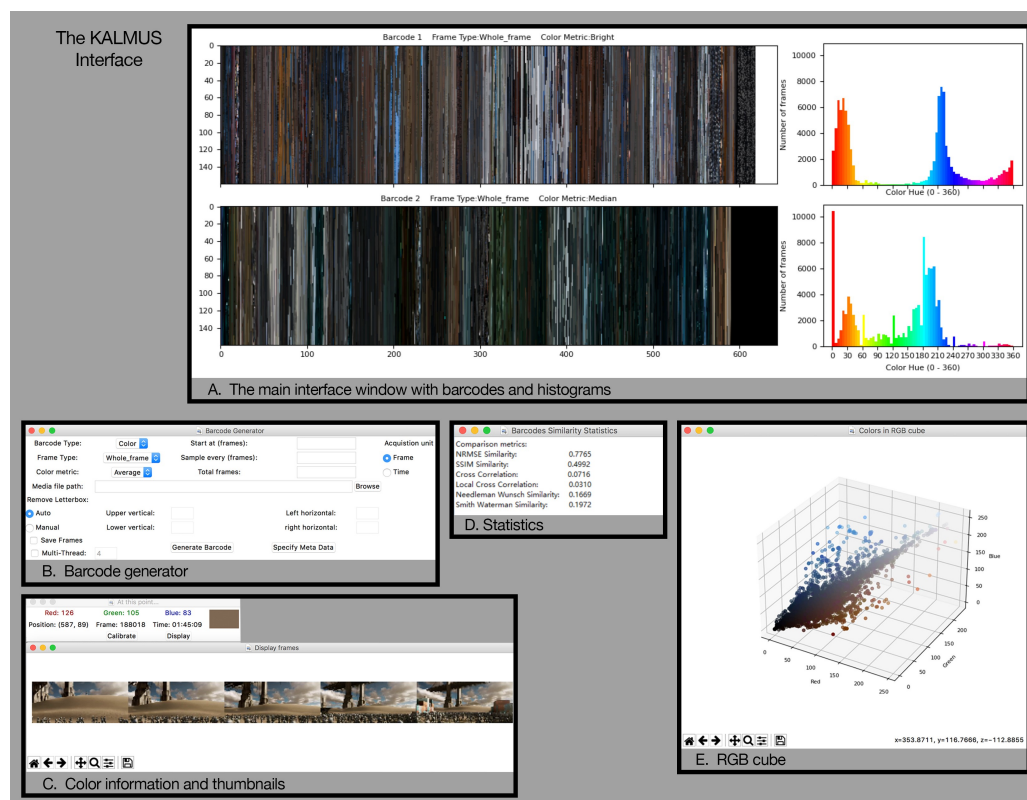
Figure 2: Image A shows barcodes for a single film in which the color of a frame was calculated in two ways: the top one uses the frame's brightest color and the bottom one uses the frame's median color. Image B shows the interface for generating a barcode: one can process a film in parallel, choose a sampling rate, and choose a method for determining a frame's color. Image C is the result of selecting a pixel in a barcode to give a user an idea of where in the film they are. Image D is a table of similarity metrics between the two barcodes. Image E is an interactive 3D plot of the RGB values in one of the barcodes.

KALMUS allows users to understand a film's color palette and compare that palette to other films. See Table 1 for options on what parts of the frame a user might want to analyze and the various metrics a user can use to determine the color of a frame. See Table 2 for information on how film colors can be compared. Potentially, films from a certain time period or a particular genre might be aggregated to see if there are common color palettes[2]. While film studies as a discipline has long depended on qualitative analysis, KALMUS provides a straightforward quantitative tool that can supplement the reading of a film. For example, in Adams et al. (in preparation) KALMUS is used to study color trends in Hollywood movies from 1990 to 2015. In a class taught by the second author, students used KALMUS to explore how a film's color palette signaled narrative shifts and introduced significant characters.

	Mean	Median	Mode	TD	WD	BR	BP
Whole frame	Yes	Yes	Yes	Yes	Yes	Yes	Yes
High contrast region	Yes	Yes	Yes	Yes	Yes	Yes	No
Low contrast region	Yes	Yes	Yes	Yes	Yes	Yes	No
Foreground	Yes	Yes	Yes	Yes	Yes	Yes	No
Background	Yes	Yes	Yes	Yes	Yes	Yes	No

---

[2]We point out users should be aware of the fact that different color gamuts used in digitizing films may affect a film's original color palette. KALMUS is designed to use the broadcast color gamut called REC 709.

Table 1: A summary of the various metrics one can use to determine the color of a frame and what parts of a frame will be used to calculate the color of the frame. The first three metrics (mean, median and mode) are the corresponding statistic over all the colors in the part of the frame being analyzed. The next two metrics, top dominant (TD) and weighted dominant (WD) are determined by applying a clustering algorithm on the set of all the colors in the part of the frame being analyzed and, in the case of the top dominant metric, the color is determined by the color of the largest cluster and, in the case of the weighted dominant metric, the color is determined by the average color of the clusters, weighted by the sizes of the clusters. The last two metrics assign the color of the region being analyzed to be the color of the brightest region (BR) or the brightest pixel (BP) in the region.

Comparison metric	Range	References
Normalized root mean square error	0 least similar, 1 most similar	Wang et al. (2004)
Structural similarity index	0 least similar, 1 most similar	Wang et al. (2004)
Cross correlation	-1 anti-similar, 1 most similar	Avants et al. (2008)
Local cross correlation	-1 anti-similar, 1 most similar	Avants et al. (2008)
Needleman–Wunsch	0 least similar, 1 most similar	Needleman & Wunsch (1970)
		Adams et al. (in preparation)
Smith–Waterman	0 least similar, 1 most similar	Smith et al. (1981)
		Adams et al. (in preparation)

Table 2: A summary of the various metrics included in KALMUS to compare the overall color of two films.

As of version 1.3.5, the software is stable. Because the audience of potential users of KALMUS is broad, we provide access to the KALMUS package via the GUI described above but also point out that the package can be installed and used in any Python environment.

Acknowledgments

References

Adams, M., Chen, Y., Faden, E., Ryan, N. C., & Zhang, H. (in preparation). *The changing colors of Hollywood films*.

Arnold, T., & Tilton, L. (2020). Distant Viewing Toolkit: A Python Package for the Analysis of Visual Culture. *Journal of Open Source Software*, *5*(45), 1800. https://doi.org/10.21105/joss.01800

Avants, B. B., Epstein, C. L., Grossman, M., & Gee, J. C. (2008). Symmetric diffeomorphic image registration with cross-correlation: Evaluating automated labeling of elderly and neurodegenerative brain. *Medical Image Analysis*, *12*(1), 26–41. https://doi.org/10.1016/j.media.2007.06.004

Flueckiger, B. (2017). A digital humanities approach to film colors. *Moving Image: The Journal of the Association of Moving Image Archivists*, *17*(2), 71–94. https://doi.org/10.5749/movingimage.17.2.0071

Flueckiger, B., & Halter, G. (2020). Methods and advanced tools for the analysis of film colors in digital humanities. *Digital Humanities Quarterly: Special Issue Digital Humanities&Film Studies. Analyzing the Modalities of Moving Images*, *14*(4).

Halter, G., Ballester-Ripoll, R., Flueckiger, B., & Pajarola, R. (2019). VIAN: A visual annotation tool for film analysis. *Computer Graphics Forum*, *38*, 119–129. https://doi.org/10.1111/cgf.13676

Itseez. (2015). *Open source computer vision library*. https://github.com/itseez/opencv.

Needleman, S. B., & Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, *48*(3), 443–453. https://doi.org/10.1016/b978-0-12-131200-8.50031-9

Smith, T. F., Waterman, M. S., & others. (1981). Identification of common molecular subsequences. *Journal of Molecular Biology*, *147*(1), 195–197. https://doi.org/10.1016/0022-2836(81)90087-5

Stutz, O. K. (2016). *Algorithmische farbfilmästhetik: Historische sowie experimentell-digitale notations-und vislualisierungssysteme des farbfilms im zeichen der igital humanities 2.0 und 3.0* [PhD thesis]. Universität Zürich.

Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, *13*(4), 600–612. https://doi.org/10.1109/tip.2003.819861

Yumibe, J. (2012). *Moving color: Early film, mass culture, modernism*. Rutgers University Press. https://doi.org/10.1080/17460654.2013.837593