

A graph-based color lines model for image analysis

D Duque-Arias, Santiago Velasco-Forero, Francois Goulette, Jean-Emmanuel Deschaud, Beatriz Marcotegui

▶ **To cite this version:**

D Duque-Arias, Santiago Velasco-Forero, Francois Goulette, Jean-Emmanuel Deschaud, Beatriz Marcotegui. A graph-based color lines model for image analysis. International Conference on Image Analysis and Processing, Sep 2019, Trento, Italy. 10.1007/978-3-030-30645-8_17 . hal-02182477v2

HAL Id: hal-02182477
https://hal.science/hal-02182477v2

Submitted on 4 Oct 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A GRAPH-BASED COLOR LINES MODEL FOR IMAGE ANALYSIS

D. Duque-Arias[1], S. Velasco-Forero[1], J-E Deschaud[2], F. Goulette[2], B. Marcotegui[1]

1: MINES ParisTech, PSL Research University, CMM-Center of Mathematical Morphology, 77300 Fontainebleau, France

2: MINES ParisTech, PSL Research University, CAOR-Centre de Robotique, 75272 Paris, France

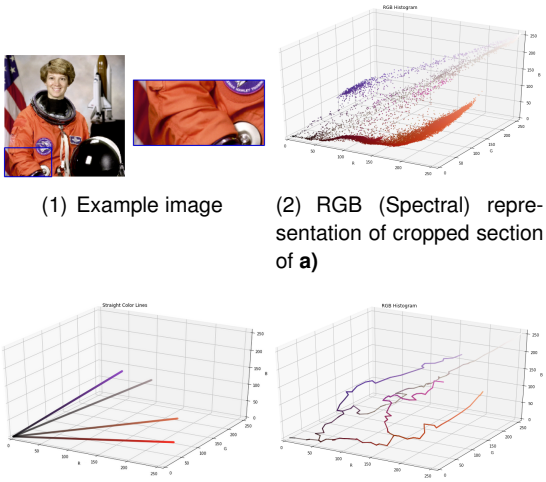
Abstract

This paper addresses the problem of obtaining a concise description of spectral representation for color images. The proposed method is a graph-based formulation of the well-known Color Lines model. It generalizes the lines to piecewise lines, been able to fit more complex structures. We illustrate the goodness of proposed method by measuring the quality of the simplified representations in images and videos. The quality of video sequences reconstructed by means of proposed color lines extracted from the first frame demonstrates the robustness of our representation. Our formalism allows to address applications such as image segmentation, shadow correction among others.

Keywords : *Color lines, Gaussian Mixture Model, minimum spanning tree, spectral representation, graph based modeling*

1. Introduction

Color is an effective cue in computer vision for identifying objects/regions of interest in images. On one hand, many efforts have been made to find an adequate color representation via linear and non-linear color spaces as RGB, HSV, and so on (1). *Color modeling*, on the other hand, plays a different role to model the color diversity as an information vector in a given representation space. These techniques include statistical models (2), color versions of level sets (3; 4), color lines (5), color-unmixing (6), lattice structures on color spaces (7), among others. In this work, we propose to model color information by piecewise linear functions, computed from a simplified version of spectral representation in the same spirit of *color lines* (5) which have demonstrated successful results in various applications as video compression and segmentation (8), interactive shadows correction (9) and image dehazing (10). Our model considers non-straight lines by computing successively longest path on a graph-based representation. Accordingly, we called our model *Graph-based Color Lines* (GCL). The proposed method is mainly divided in three steps: **i)** Compute simplified spectral representation; **ii)** Graph modeling of the color representation; **iii)** Build graph-based color lines. We note that third step is a graph-based generalization of color lines method (5), based of the spectral representation of images. Our GCL model, allows to identify more complex geometrical structures in spectral representations than Straight Color Lines (SCL) (8) but keeping at same time the potential to analyze the image. Additionally, GCL are not constrained to have orientation in the positive directions as it is the case of classical CL (5). We highlight that motivation behind our contribution is to obtain a reproducible model and preserving the versatility of original color line model (5). As a manner of example, we compare the results obtained by the proposed method against SCL (8), in the example illustrated in Fig. 1. For this case, four color lines were extracted for both representation and a quantitative measure is reported (34.96 dB for GCL compared to 21.66 dB on the SCL) by means of PSNR for a reconstructed image. Details of the computation are presented in Section 3.



(1) Example image

(2) RGB (Spectral) representation of cropped section of **a)**

(3) Straight Color Lines (PSNR: 21.66 dB)

(4) Graph-based Color Lines (PSNR: 34.96 dB)

Figure 1: Comparison between SCL and GCL. Our model fits spectral representation by piecewise lines.

The rest of the paper is organized as follows: in Section 2, we describe the proposed methodology and mathematical fundamentals for computing the proposed GCL model. In Section 3, some experiments are presented to compare SCL and our GCL for some images and videos. We conclude our findings in Section 4.

2. Graph-based Color Lines Model

The proposed methodology for computing our Graph-based Color Lines (GCL) is divided in three stages. As

descriptive example, the image in Fig. 1(a) was used to illustrate all required intermediate steps in Figs. 3, 4 and 6.

2.1. Simplified color representation

The aim of the first stage is to find a simplified spectral representation of the input image. We follow the same approach proposed by (5) and it is summarized in the workflow of Fig. 2.

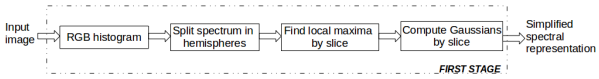


Figure 2: Steps for computing simplified spectral representation.

Firstly, the RGB histogram is computed and associated pixels to each color are stored. It is important to highlight that: **i)** Each point of the spectral representation belongs to a different color in the image. In this work, we call it "color point"; **ii)** A color point may be associated to more than one pixel from the image; **iii)** Color points located further from the origin have a greater luminance. Fig. 1(b) shows spectral representation from the image(a). There are visually identifiable geometric structures representing related colors in the image. This kind of structures are common in real world images as stated in (11). From obtained spectral representation, the RGB space is divided in equally separated hemispheres to group the color points according to its luminance, as proposed by (5), to preserve the spectral diversity. The criteria to verify if a color point x is contained in a slice between two consecutive hemispheres with radius r_i and r_j, is given by $r_i \leq \|x\| < r_j$. Following the original formulation of color lines model (5), we fit a Gaussian mixture model (GMM) by hemisphere where the number of components is given by the number of local maxima. It allows to consider color points associated to colors with high density in the image. Additionally, as proposed by (5), the center of each Gaussian distribution is initialized at each local maximum, removing the instability of random initialization in GMMs. Due to spatial split in hemispheres, GMMs describe local behavior of pixels on the slice and global spectral diversity of the image is preserved.

Image (a) of Fig. 1 contains 13.851 color points. After the calculation of the local maxima by slice, the number of Gaussian distributions that describes the spectral representation is $n = 313$ (see Fig 3).

2.2. Graph modeling of color representation

In the second stage, the input is a simplified representation of the image given by the matrix $\mathbf{M}$, defined as $\mathbf{M} = [\mu_0, \mu_1, ..., \mu_n, \mathbf{0}]$, where μ_i is the mean vector of the *i-th* Gaussian distribution, n is the number of distributions and $\mathbf{0}$ is the origin point vector of the color space. The expected output is a tree describing the geometric structures of color points. An approach to generate the

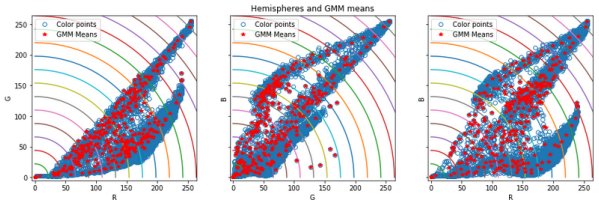


Figure 3: Gaussian means of spectral representation of cropped section Fig. 1 (a).

color lines from $\mathbf{M}$ is to assume a linear behavior in geometric structures of color points as in (8). In doing so, each color line may be built as a SCL intersecting the origin. Fig. 4 shows (calculated) SCL from Fig. 1. It is seen that obtained color lines do not fit geometric structures of the spectral representation.

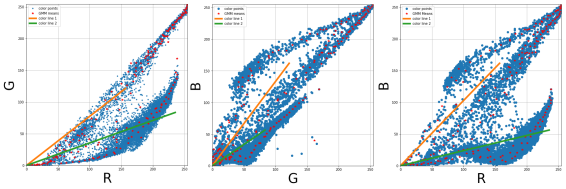


Figure 4: First two SCLs of cropped section Fig. 1 (a).

We propose to generalize the method proposed by (5) to piecewise lines, able to represent more complex geometric structures of spectral representation. In detail, the first step is to calculate the complete graph $\mathcal{G}(\mathbf{M}, \mathbf{E})$ given by: $\mathbf{M}$ each node is a vector mean of the Gaussian distributions, and $\mathbf{E}$ as the matrix of Euclidean distances between the vectors in $\mathbf{M}$. Accordingly, we proposed a graph-based approach by linking Gaussian distributions in an economic way via a Minimum Spanning Tree (MST). A MST is a subgraph of $\mathcal{G}$ that satisfies following conditions: **i)** It spans all the vertices; **ii)** Is acyclic; **iii)** Minimizes the sum of weights of its edges (12). The use of MST at this point, is motivated for different links between density estimation and the MST from Euclidean distances (13) (14).

2.3. Building graph-based color lines

In the third stage, we build a graph-based representation from color information. We note that the MST represents the spectral diversity of an image, in the spirit of color lines(5). However, in our case, we define the color lines in the MST, as the path from the origin to each one of the leaves of the tree (nodes of degree one). We call each one of these path a *color line* induced by the MST. For some applications, one can be interested in the selection of most important *color lines*. Accordingly, we propose to rank obtained lines by their length, measured as sum of their edge weights (Euclidean distance between corresponding nodes). A simple algorithm to find the top-K lines by means of Dijkstra's algorithm (12) is given in Alg. 1.

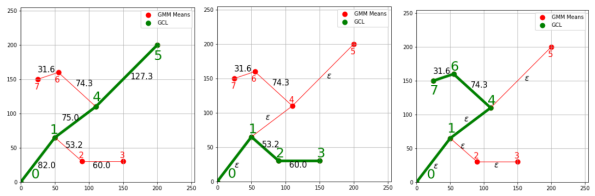
Fig. 5 shows a simple example of the implementation of Alg. 1 for computing top-3 GCL as follows: **i)** In

Input : $\mathbf{M}$ (Defined on Section 2.2)

Compute the MST of $\mathcal{G}(\mathbf{M}, \mathbf{E})$;
Add $\mathbf{0}$ to L_1;
for $k \leftarrow 1$ **to** K **do**
  Find longest path p from node $\mathbf{0}$ (Dijkstra algorithm);
  for *each edge (i,j) in* p **do**
    `//If not visited yet`
    if $\mathbf{E}(i,j) > 0$ **then**
      $\mathbf{E}(i,j) = 0$;
      Add (j) to L_k;
    end
  end
end
Output: $L_1, L_2, \ldots, L_K$ the top-K color lines

Algorithm 1: Compute top-K color lines

Fig. 5(a), the longest path from the origin is composed by $p = [0, 1, 4, 5]$ and so the first color line $L_1 = [0, 1, 4, 5]$; **ii)** After updating weights in Fig. 5(b), the longest path from the origin is composed by $p = [0, 1, 2, 3]$ and so $L_2 = [2, 3]$; finally, **iii)** After updating weights in Fig. 5(c), the longest path from the origin is composed by $p = [0, 1, 4, 6, 7]$ and so $L_3 = [6, 7]$; the assumption of assigning common nodes to longest line has some advantages: **i)** Every color point is clustered; **ii)** Darkest color points (close to the origin) are grouped on first nodes of the longest color line. According to the application where GCL is used, assigning common nodes to a particular line may affect performance. For example, as it is seen in Fig. 1 (b), color points close to the origin represents dark colors or shadowed regions. This may mislead to identifying pixels as shadowed sections of the image instead of the actual colors.



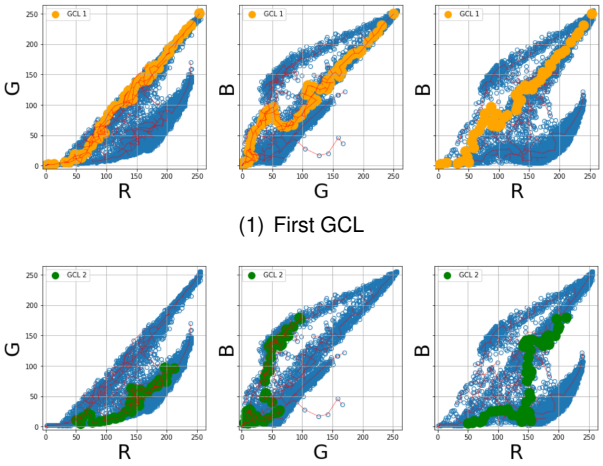
(1) First GCL (2) Second GCL (3) Third GCL

Figure 5: Illustrative example of Algorithm 1

Fig. 6 shows the two longest GCL of the cropped section of Fig. 1 (a) (it is seen how they adapt to spectrum geometry).

3. Experimental Results

The purpose of this section is to present the potential of our proposed GCL model. In order to evaluate the quality of image representation using GCL, we measured the error in the representation by mean of a projection into the model and then we are able to compare different methods *via* standard fidelity measures of representations as Peak Signal to Noise Ratio (PSNR) and Mean Squared Error (MSE).



(1) First GCL

(2) Second GCL

Figure 6: First two GCL of cropped section Fig. 1 (a)

3.1. Projection into GCL

In order to evaluate the quality of image simplification using our model, we propose to project the original image I_i into obtained GCL and measure the error of the reconstructed image. The procedure for projecting spectral representation is shown in Fig. 7:

1. *Interpolate GCL:* As presented in Section 2.3, each GCL is defined as a sequence of Gaussians of related colors. We calculated piecewise linear interpolation between means of the Gaussians belonging to the same GCL, to add soft color transitions over the line.
2. *Project original image I_i into GCL:* From Fig. 6, it can be seen that each GCL partially simplifies the color representation. In order to find the closest GCL for each color point, it is calculated Euclidean distance between each color point and every GCL. Then, it is re-assigned to each color point the value of the closest GCL. As a result, color diversity from original representation (I_i) is simplified (I_p) according to number of GCL.

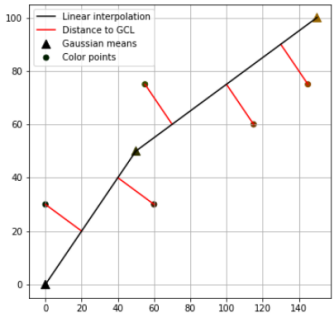


Figure 7: Illustrative example of color points projection on a GCL

For quantitative comparisons, Table 1 presents the results of PSNR and MSE using SCL and GCL. It is observed that: **i)** PSNR is above 30 dB for images using GCL. The quality of our representation is greater than the one obtained using SCL; **ii)** Images 1 and 3 have higher contrast and more different colors. Increasing the number of color lines allows a better fit to more complex geometrical structures. An example of fitting a complex structure is shown in Fig. 6, specially in the second line; **iii)** Images 2 and 4 do not present considerable variations on PSNR and MSE when the number of color lines increases. It occurs because the spectral representation of the image has few geometric structures so few color lines are required to represent the image. Table 2 shows percents of pixels grouped in each GCL. From obtained results, is found that in some images (1, 3 and 4), the percentage of pixels grouped in first four lines is over 85%.

Id	Image	K	PSNR (dB)		MSE	
			SCL	GCL	SCL	GCL
1		3	20.19	17.68	1864	3328
		4	21.65	34.96	1333	62
		5	21.65	35.14	1333	59
2		3	21.44	33.54	1380	86
		4	21.56	34.10	1361	75
		5	21.57	34.11	1366	75
3		3	27.93	30.56	314	171
		4	28.56	35.96	271	49
		5	28.70	35.99	262	49
4		3	20.29	30.51	1822	173
		4	20.34	32.63	1800	106
		5	20.36	32.65	1794	105

Table 1: PSNR and MSE. SCL vs GCL

Id	% of pixels by line				
	L_1	L_2	L_3	L_4	Sum
1	8.21	17.75	1.6	57.86	85.41
2	69.45	1.96	0.01	1.79	73.2
3	76.58	0.19	7.81	3.5	88.07
4	86.71	5.27	0.82	1.03	93.83

Table 2: Percentage of pixels by GCL.

Fig 8 displays the relationship between the number of hemispheres (varying from 1 to 40) against number of GCL (varying between 1 and 20), of full Astronaut image from Fig. 1(a). As it is seen on the top of Fig 8, a reduced number of GCL and hemispheres represents the lowest image quality. Nevertheless, increasing both (lines and hemispheres) does not allow to indefinitely improve the quality of the simplification of the image.

To show the *capacity of generalization* (15) of our method, we performed an experiment measuring PNSR error for a set of images in a video, using GCL model computed uniquely from the first frame denoted by I_0. Note that an excellent model fitting does not guarantee good performance in predicting future frames, because of possible over-fitting on I_0. We have used two videos

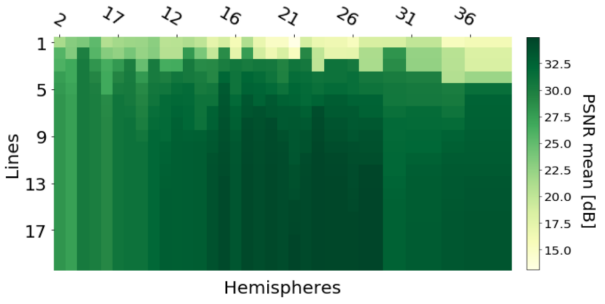
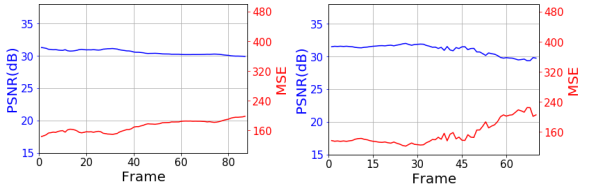


Figure 8: PSNR varying the number of GCL and hemispheres from Table 1.

from (16) (*boxing-fisheye* and *mallard-fly*) and calculated the **top-5 GCLs** on the first frame (I_0). Fig. 9 shows PSNR and MSE over the frames of each video.



(1) Boxing-fisheye (2) Mallard-fly

Figure 9: PSNR and MSE in two videos from (16). PSNR (blue) and MSE (red).

We observe that PSNR and MSE were stable in the *boxing-fisheye* video. It occurs because the video corresponds to a sequence of frames without significant changes. In the video *mallard-fly*, the error increases for the last frames. It happens because there is camera panning and the last frames have nothing in common with I_0, for which GCL were calculated. More examples of our methods for different videos of (16) are available at `http://www.cmm.mines-paristech.fr/~duque/gcl`

Figs. 10 and 11 show the first frame from videos of Fig. 9. It is seen how the color diversity changes according to the number of GCL in the model. Even though, as is presented in Fig. 8, adding GCL to the model does not guarantee an indefinitely gain in PSNR.

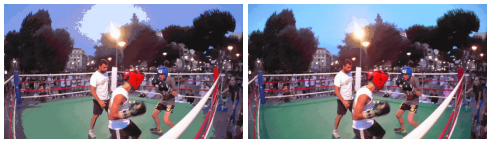
4. Conclusions and Future Work

In this paper, we present a graph-based method to model spectral diversity of RGB images in the spirit of Color Lines(5). The resulting GCL model performs better than (8) in most of the explored examples, which is natural due to the higher complexity of our model (piecewise linear function instead of straight lines). However, experiments on videos show that GCL model computed on a single frame does not over-fit and describes correctly the whole sequence, within the same scene. Finally, we noted that GCL is general-purpose model and we are planning to use it for image/video segmentation, image/video simplification, change detection, etc. In future research, we will also work on methods to correct

shadows and dehazing images using GCL instead of CL as in (9; 10).



(1) Original image (2) 1 GCL (19.74 dB)



(3) 5 GCL (24.06 dB) (4) 20 GCL (30.97 dB)

Figure 10: Images from boxing-fisheye (16).



(1) Original image (2) 1 GCL (24.2 dB)



(3) 5 GCL (29.81 dB) (4) 20 GCL (31.67 dB)

Figure 11: Images from mallard-fly (16).

Références

[1] Konstantinos N Plataniotis and Anastasios N Venetsanopoulos. *Color image processing and applications*. Springer Science & Business Media, 2013.

[2] Haim Permuter, Joseph Francos, and Ian Jermyn. A study of gaussian mixture models of color and texture features for image classification and segmentation. *Pattern Recognition*, 39(4):695–706, 2006.

[3] Daniel Cremers, Mikael Rousson, and Rachid Deriche. A review of statistical approaches to level set segmentation: integrating color, texture, motion and shape. *International Journal of Computer Vision*, 72(2):195–215, 2007.

[4] Michèle Gouiffès and Bertrand Zavidovique. Body color sets: A compact and reliable representation of images. *Journal of Visual Communication and Image Representation*, 22(1):48–60, 2011.

[5] Ido Omer and Michael Werman. Color lines: Image specific color representation. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–II. IEEE.

[6] Yağız Aksoy, Tunç Ozan Aydın, Aljoša Smolić, and Marc Pollefeys. Unmixing-based soft color segmentation for image manipulation. *ACM Trans. Graph.*, 36(2):19:1–19:19, 2017.

[7] Jesús Angulo. Morphological colour operators in totally ordered lattices based on distances: Application to image filtering, enhancement and analysis. *Computer Vision and Image Understanding*, 107(1-2):56–73, 2007.

[8] T. Nishikawa and Y. Tanaka. Dynamic color lines. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 2247–2251, Oct 2018.

[9] Xiaoming Yu, Ge Li, Zhenqiang Ying, and Xiaoqiang Guo. A new shadow removal method using color-lines. pages 307–319, 07 2017.

[10] Raanan Fattal. Dehazing using color-lines. volume 34, pages 1–14, New York, NY, USA, 2014. ACM.

[11] Antoni Buades, Jose Luis Lisani, and Jean-Michel Morel. On the distribution of colors in natural images. 2010.

[12] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to algorithms*. MIT press, 2009.

[13] Sreevani and C.A. Murthy. On bandwidth selection using minimal spanning tree for kernel density estimation. *Computational Statistics Data Analysis*, 102:67 – 84, 2016.

[14] Zhiding Yu, Oscar C Au, Ketan Tang, and Chunjing Xu. Nonparametric density estimation on a graph: Learning framework, fast approximation and application in image segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2201–2208. IEEE, 2011.

[15] Kenji Kawaguchi, Leslie Pack Kaelbling, and Yoshua Bengio. Generalization in deep learning. *arXiv preprint arXiv:1710.05468*, 2017.

[16] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Computer Vision and Pattern Recognition*, 2016.

Acknowledgements

This work was partially funded by REPLICA FUI 24 project and ARMINES.