sciendo

TATRA MOUNTAINS
Mathematical Publications
DOI: 10.2478/tmmp-2018-0019
Tatra Mt. Math. Publ. **72** (2018), 31–41

COMPARATIVE ASSESSMENT OF THE IMAGE DIVIDE AND LINK ALGORITHM IN DIFFERENT COLOR SPACES

Carely Guada[1] — Daniel Gómez[2] — J. Tinguaro Rodríguez[1] — Javier Montero[1,3]

[1]Universidad Complutense de Madrid, Fac. de Ciencias Mat., Madrid, SPAIN

[2]Universidad Complutense de Madrid, Fac. de Estudios Estadísticos, Madrid, SPAIN

[3]Universidad Complutense de Madrid, Inst. IGEO CSIC-UCM, Madrid, SPAIN

ABSTRACT. In this paper, a comparative assessment of the Image Divide and Link Algorithm (ID&L) in different color spaces is presented. This, in order to show the significance of choosing a specific color space when the algorithm computes the dissimilarity measure between adjacent pixels. Specifically, the algorithm procedure is based on treating a digital image as a graph, assigning a weight to each edge based on the dissimilarity measure between adjacent pixels. Then, the algorithm constructs a spanning forest through a Kruskal scheme to order the edges successively while partitions are obtained. This process is driven until all the pixels of the image are segmented, that is, there are as many regions as pixels. The results of the algorithm which have been compared with those generated using different color spaces are shown.

1. Introduction

Image processing has the main objective of solving complex tasks, for which it uses methods or techniques such as image capture, digitization, pre-processing, edge detection, region segmentation, classification of regions, high-level identification, quantitative conclusions/qualitative, among others [18]. In this way, depending on the task to be developed and the desired output, it is necessary

2010 Mathematics Subject Classification: 68XX, 68U10.
Keywords: edge detection, color spaces, hierarchical partition, graph-based approach.
This research has been partially supported by the Government of Spain (grant TIN2015--66471-P), the Government of Madrid (grant S2013/ICE-2845, CASI-CAM-CM), Complutense University (UCM research group 910149).

to apply a certain technique. In other words, there is a variety of image processing techniques aimed at solving certain types of tasks that are executed using different techniques and algorithms of information processing and automatic learning. For example, edge detection, as the name suggests, detects edges in an image by analyzing the intensity function of each pixel [3], [18]. On the other hand, segmentation is a technique in which an image is divided into multiple regions or objects, in order to simplify or modify the representation of the image to make it more meaningful and easy to analyze [1]. In addition, sometimes it is necessary to combine different techniques to achieve the desired objective.

Furthermore, a digital image can be associated to a graph, where the nodes represent the pixels of the image and the edges represent the relations between pairs of pixels. Thus, the digital image can be modeled by a two-dimensional structure of pixels, where each pixel is connected to its natural neighborhood in the image [7]. In this way, some image processing techniques are based on graph theory, for example normalized cuts [16], intelligent scissors using Dijkstra algorithm [13], graph theory approach and its connection to Markov fields [11], random walks [20], graph cuts [2] and regions adjacency graphs [22].

In addition, color spaces in image processing plays an important role because in most cases it helps in the simplification and extraction of objects in a digital image. Color spaces can be represented by a limited number of dimensions and are usually defined as geometrical models of human color experiences [14]. Most models try to encompass the greatest number of colors visible to the human eye, although many times they try to isolate a set of them with some purpose. Some of the most known color spaces in image processing are RGB, HSV, YUV, CIELab. Table 1 shows the color space and its number of dimensions.

Table 1. Color space and its dimensions.

Dimensions	Color space
1	Gray
2	RG, HS, a∗b
3	RGB, CMY, HSV, YUV, CIE, XYZ
4	CMYK

In summary, the objective of a color model is to facilitate the specification of colors in a standardized and generally accepted way. Thus, a color model refers to the specification of a three-dimensional coordinate system and a subspace of this system in which each color is represented by a single point.

In this paper, the performance of the edge detection algorithm ID&Ln based on a graph theory approach, is compared on different color spaces. Edge detection is one of the most important research areas of image processing and its applications.

Edge detection methods identify the edges of a digital image through analysing the brightness variations between the image pixels. In more detail, edge detection methods analyze the intensity function of each pixel to determine a set of edges when a brightness changes sharply occurs. There are some algorithms for edge detection (see for example [3], [17]). An important aspect of these algorithms is that the image input has to be in gray-scale image.

However, there is an edge detection algorithm based on a hierarchical graph-partition approach that can work with a color image as an input. This algorithm is called *Image-Divide-and-Link* (ID&L) [8]. The ID&L algorithm consists on a divisive method that allows a hierarchical partition of a network with a low computational cost and then, identifies the edges over the hierarchical partitions.

The structure of this paper is following: in Section 2, we provide a basic description of color spaces. In Section 3, an edge detection algorithm based on a hierarchical clustering named Image Divide and Link is described. Section 4 contains a comparison of all four color spaces considered over the Divide and Link algorithm. Finally, the conclusions are summarized in Section 5.

2. Color space

A digital image can be represented in different color space for several applications of image processing [4], [5]. A color space is a system of color interpretation, that is, a specific organization of colors in an image or video. A color space can be arbitrary, with particular colors assigned according to the system and mathematically structured.

In next subsection, some different color spaces are described.

2.1. Categorization of color spaces

Color spaces can be classified in the following way:

RGB based color space: is based on three components, red (R), green (G) and blue (B), describing a color space with additive color composition [9].

HSV based color space: is based on three components, hue, saturation and brightness and is very useful because it is often more natural since it separates out the intensity from the color information [19].

YUB based color space: is based on the characteristics of the human visual perception, describing a color by using the color components luminance and chrominance [9].

CIE color spaces: two of their properties are device-independency and perceptual linearity. Some of these types are CIE XYZ, Lab and Luv [21].

Notice that in general a color of a certain color space can be converted to another color space, and vice versa. However, some color spaces may have range limitations, and the conversion of colors outside that range will not produce correct results. In addition, it is possible that there might be rounding errors when converting, especially when the range of 256 different values is maintained for each dimension.

Next section is dedicated to present the ID&L edge detection algorithm, which will be applied on different color spaces.

3. Image Divide and Link algorithm

The Image Divide and Link algorithm (D&L) [8] is an edge detection algorithm based on a hierarchical graph-partition approach [7]. Before we introduce how the algorithm works, we define some basic concepts.

3.1. Digital image representation

A digital image $I_{M\times N}$ can be represented by a graph whose nodes are the pixels and the edges expressed the neighbourhood between the pixels. Formally, let $G=(V,E)$ be a graph of a digital image $I_{M\times N}$, let

- $V=\left\{p_{i,j} | 1 \leq i \leq r; 1 \leq j \leq s\right\}$ be the set of pixels;
- $E=\left\{e=\{p_{ij};p_{i'j'}\} | p_{ij},p_{i'j'} \in V\right\}$ be the set of non-ordered pairs of neighbour pixels.

There exists an edge $e=\{p;p'\}\in E$ if two pixels p and p' are neighbours, otherwise $e=\{p;p'\}\nexists E$. A simple topology for an image is the one that a pixel is linked with its four neighbours. The four neighbours topology is used in this work for simplicity (see Figure 1).

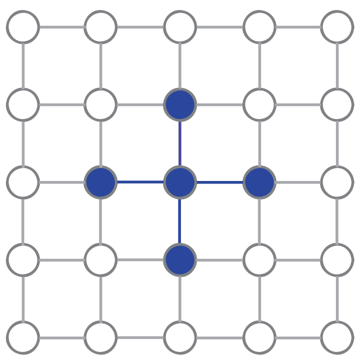

FIGURE 1. Four neighbours topology.

Each $e = \{p; p'\} \in E$ edge has a weight which expresses the degree of dissimilarity ($d_e \geq 0$) between two pixels, where the greater d_e is, the more dissimilar p and p' are. Thus, let $D = \{d_e | e \in E\}$ be the set of all dissimilarity between neighbours pixels. Notice that establishing the degree of dissimilarity ($d_e \geq 0$) depends on the problem under consideration. In this way, the complete information about a digital image $I_{M \times N}$ can be summarized by a network: $N(I_{M \times N}) = \{G; D\}$ (see Figure 2).

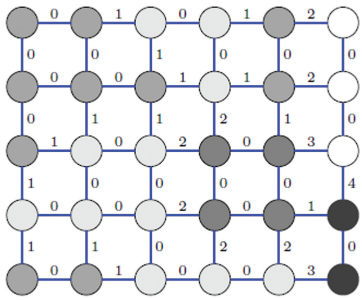

Figure 2. Network N.

Image Divide and Link (ID&L) algorithm consists in a divisive process which starts from a unique cluster and ends with partitions obtained successively until having as many clusters and pixels as an image network [8].

Before introducing the ID&L algorithm, some definitions of graph partition context are introduced.

3.2. Graph partition

Given $G = (V, E)$ a graph, let P be a graph partition and let $P = \{C_1, \ldots, C_r\}$ be a set of partition, it holds that:

- $\bigcup_{j=1}^{r} C_j = V$, covering.
- $C_i \cap C_j = \emptyset \; \forall \; i \neq j$ (disjoints).
- $\forall i$ the subgraph $(C_i, E_{|C_i})$ is connected.

Before we introduce the hierarchical partition concept, first we define the finer partition concept. Let P and Q be two partitions of a graph $G = (V, E)$. We say that P is finer than Q ($P \tilde{\subseteq} Q$) if $\forall A \in P$ there is $B \in Q$ such that $A \subseteq B$.

Then, the hierarchical partition concept is presented. Let $G = (V, E)$ be a graph and let $D = (P^0, P^1, P^2, \ldots, P^r)$ be a sequence of partitions. We say that D is a hierarchical segmentation of G (obtained in a divisive form) if:

- There exist two trivial partitions: P^0 the partition which has all pixels; and P^r containing all pixels as singleton clusters $P^r = \big\{\{i\}, i \in V\big\}$.

- $|P^i| > |P^{i-1}|$ $\forall i = 1, \dots, r$, (in each iteration the number of cluster increase).
- $P^i \tilde{\subseteq} P^{i-1}$ $\forall i = 1, \dots, r$, P^i is finer than P^{i-1}.

3.3. Graph coloring

A graph coloring induces a partition of a graph. Let $i \in V$ be a binary colored node ($\mathrm{col}(i) \in \{0, 1\}$). Then the color of any adjacent node $j \in V$ depends on the dissimilarity measure d_e of each $e = \{i, j\} \in E$ when it is compared with a predefined threshold α:

$$\mathrm{col}(j) = \begin{cases} \mathrm{col}(i), & \text{if } d_e < \alpha, \quad \forall e \in E, \\ 1 - \mathrm{col}(i), & \text{if } d_e \geq \alpha, \quad \forall e \in E. \end{cases}$$

In order to color a graph, it has to be over a *spanning forest* related with the graph under consideration, to avoid cycles. A spanning forest or spanning tree is a graph without cycles, and their connected components are trees. Given a N network and a set of thresholds $\alpha_1 > \alpha_2 > \cdots > \alpha_K$ where $\alpha_1 = \max_{e \in E}\{d_e\}$ and $\alpha_K = \min_{e \in E}\{d_e\}$, let $\{G^i = (V, E^i)\}$ with $i \in \{1, 2, \dots, K\}$ and $E^i = \{e \in E | d_e \geq \alpha_i\}$. Let us build the family of spanning forest $F^i = (V, T^i)$ through the following steps:

- Let $F^i = (V, T^i)$ the minimum spanning forest of the partial graph
  $$G^i | G^{i+1} = (V, E^i - E^{i+1}).$$
  Following a similar Kruskal algorithm [10], the construction of F^i is based on sorting the edges whose weights belong to the interval $(\alpha_i, \alpha_{i+1}]$ in a decreasing order.
- If F^i is already a spanning forest of $G^i | G^{i+1}$, end the process. Otherwise, edges lower than α_i will be sorted in increasing order, and then they are iteratively added to T^i (if they do not make cycles); and the process is finished when F^i is a spanning forest.

3.4. Contraction of nodes

In order to maintain compact homogeneous regions for any threshold in the coloring procedure, we need to consider a similarity threshold α_τ. Assuming a similarity threshold α_τ, any edge lower than this value and those which were incident to them are considered as a unique pixel. In this way, these pixels will be never separated in the process. Let $E(\alpha_\tau) = \{e_{ab} \in E | d_{ab} \leq \alpha_\tau\}$ be the set of edges with lower weights than this value. Then, any connected component of the $G(\alpha_\tau) = (V, E(\alpha_\tau))$ partial graph is merged into a single node.

In summary, ID&L partitions the set of pixels of the $G = (V, E)$ graph by a binary iterative procedure that classifies V nodes in two classes, V_0 and V_1. At any step, $E_d \subset E$ represents the set of edges of division and $E_l \subset E$ represents the set of link edges.

- If $e = \{p; p'\} \in E_d$, then the nodes p and p' are classified in different classes (e.g., $p \in V_0$ and $p' \in V_1$).
- If $e = \{p; p'\} \in E_l$, then the nodes p and p' are classified in the same class (e.g., $p, p' \in V_0$).

3.5. Determining boundary maps

Once the hierarchical region map is determined, we delimit the edges of each region. To this aim, the pixels of a given hierarchical region map are classified into *White* and *Black* classes as follows: Let $G = (V, E)$ be a graph, and let $\mathcal{P} = \{C_1, \ldots, C_t\}$ be a partition of such a graph. Let also $C(p) \in \mathcal{P}$ denote the region to which the pixel p belongs, with $p \in V$ for all $i \in \{1, \ldots, n\}$. Then, the *White* and *Black* classes are defined in the following way:

- $Black = \big\{p \in V \mid C(p) = C(p) \; for \; all \, e_{ij} = (p, p') \in E\big\}$.
- $White = V - Black$.

3.6. General algorithm

The ID&L algorithm consists of the following main steps:

(1) Calculate the weights of division and link to each edge.
(2) Organize subgraph edges G^t sequentially according to the weights of the previous step.
(3) Build a support forest $F^t \subset G^t$ (similar to Kruskal algorithm [10]) based on the arrangement found it in the previous step.
(4) Build partition P^t following the binary procedure applied to F^t.
(5) Define new set of edges E^{t+1} from E^t by removing those edges that connect different groups of P^t.
(6) Repeat steps 1-5 while $E^{t+1} \neq \emptyset$.

3.7. Example of the algorithm Image Divide and Link

In order to visualize an output of the ID&L algorithm, let us consider the image shown in Figure 3, that has a size of 478×321 pixels.

The output of the ID&L algorithm is shown in Figure 4. As it can be seen, an advantage of the ID&L algorithm is that it provides the sequence of partitions obtained. In this way, the user has the option to choose the part of the sequence that is most informative to solve its problem, contrarily to other algorithms that provide a single partition image.

We would like to remark that since the ID&L algorithm modelled a digital image as a graph, it can be used in images with more dimensions, as for instance, hyperspectral images.

Figure 3. Original image.

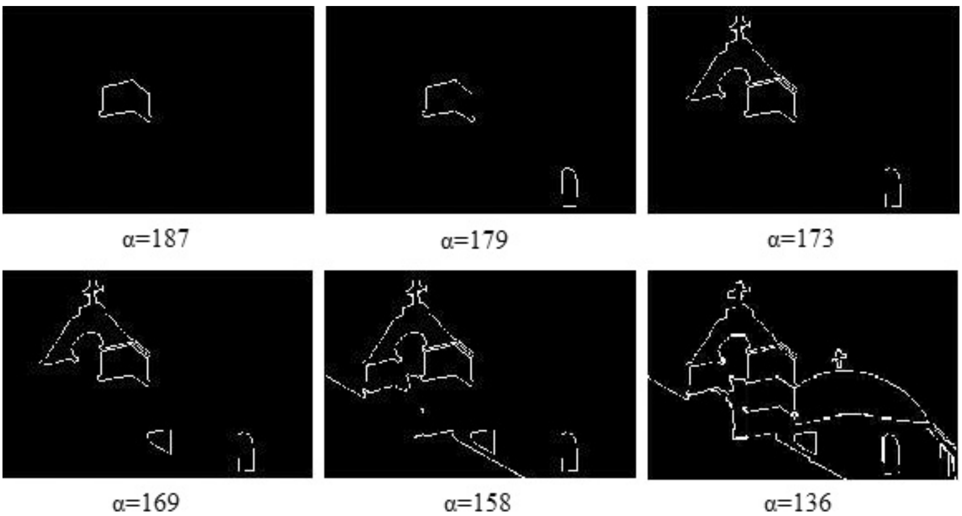

Figure 4. D&L output.

4. Computational experiences

In this section, the assessment of the ID&L algorithm is performed by comparing the outputs obtained in different color spaces, RGB, CIELab, YUV and HSV. To this aim, we applied the ID&L algorithm over 20 images of the Berkeley Segmentation Dataset (BSDS500) [12]. First, we applied a pre-processing over the input image (a RGB image) to convert it to each of the color spaces considered.

Once the pre-processing was realized, we applied the ID&L algorithm on those images, including the RGB image. The dissimilarity measure considered between every channel or dimension is the Euclidean distance. Let $P_1^{\mathrm{dcs}} = (d_1, c_1, s_1)$ and $P_2^{\mathrm{dcs}} = (d_2, c_2, s_2)$ be two pixels on a specific color space and "dcs" represents each dimension of the color space. Then, to measure the dissimilarity between P_1 and P_2 the following expression is used:

$$d_{1,2}^{\mathrm{dcs}} = \sqrt{(\Delta d)^2 + (\Delta c)^2 + (\Delta s)^2},$$

where

$$\Delta d = d_2 - d_1, \quad \Delta c = c_2 - c_1 \quad \text{and} \quad \Delta s = s_2 - s_1.$$

In addition, we considered a set of thresholds of $K = 7$.

Finally, in order to quantify the performance of each color space over the ID&L, we computed the F-measure [6] on each output of the ID&L following the approach of [15]. The resulting F-measures are shown in Table 2.

TABLE 2. F-measure results.

Color Space	F-measure		
	max	mean	min
HSV	0.5472	0.4743	0.3432
YUV	0.5781	0.5066	0.3767
CIELab	**0.6081**	**0.5337**	**0.3975**
RGB	0.5372	0.4663	0.3386

5. Conclusions

An advantage of the ID&L algorithm is that it provides a sequence of partitions. In this way, the user has the option to choose the part of the sequence that is most informative. Contrary to other algorithms that provide a single partition output created from any constraint and that output may not be so informative.

As we have seen, the selection of the color space is very important in image processing, specifically in algorithms that work with a dissimilarity measure. The Image Divide and Link algorithm has better performance using the CIELab color space. Indeed, the CIELab color space is the chromatic model normally used to describe all the colors that the human eye can perceive.

REFERENCES

[1] BASAVAPRASAD, B.—RAVINDRA, H.: *A survey on traditional and graph theoretical techniques for image segmentation.* In: IJCA Proceedings on National Conference on Recent Advances in Information Technology—NCRAIT '14, pp. 38–46.

[2] BOYKOV, Y.—FUNKA-LEA, G.: *Graph cuts and efficient n-d image segmentation,* Int. J. Comput. Vis. **70** (2006), 109–131.

[3] CANNY, J.: *A computational approach to edge detection.* In: IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-**8** (1986), 679–698.

[4] CHEN, W.—SHI, Y. Q.—XUAN, G.: *Identifying computer graphics using hsv color model and statistical moments of characteristic functions.* In: IEEE International Conference on Multimedia and Expo—ICME '07, Beijin, China, 2007, IEEE, pp. 1123–1126.

[5] CHENG, H.-D.—SUN, Y.: *A hierarchical approach to color image segmentation using homogeneity*, IEEE Trans. Image Process. **9** (2000), 2071–2082.

[6] ESTRADA, F. J.—JEPSON, A. D.: *Benchmarking image segmentation algorithms*, Int. J. Comput. Vis. **85** (2009), 167–181.

[7] GÓMEZ, D.—ZARRAZOLA, E.—YÁÑEZ, J.—MONTERO, J.: *A divide-and-link algorithm for hierarchical clustering in networks*, Inform. Sci. **316** (2015), 308–328.

[8] GUADA, C.—ZARRAZOLA, E.—YÁÑEZ, J.—RODRÍGUEZ,J. T.—GÓMEZ, D.– –MONTERO, J.: *A novel edge detection algorithm based on a hierarchical graph-partition approach*, J. Intell. Fuzzy Syst. **34** (2018), 1875–1892.

[9] KERMIT IMAGE TOOLKIT: *Color formats for image and video processing*, `www.equasys.de/colorformat.html,2017`.

[10] KRUSKAL, J.: *On the shortest spanning subtree of a graph and the traveling salesman problem*, Proc. Amer. Math. Soc. **7** (1956), 48–50.

[11] LI, S. Z.: *Markov random field models in computer vision.* In: 3rd European Conference on Computer Vision—ECCV '94 (J.-O. Eklundh, ed.), Stockholm, Sweden, 1994, Lecture Notes in Comput. Sci., Vol. 801, Springer–Verlag, Berlin, 1994, pp. 361–370.

[12] MARTIN, D.—FOWLKES, C.—TAL, D.—MALIK, J.: *A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics*, in: Proc. of the 8th Internat. Conf. in Computer Vision—ICCV '01, Vol. 2 (B. Werner, ed.), Vancouver, BC, Canada, 2001, IEEE, pp. 416–423.

[13] MORTENSEN, E.—BARRETT, W.: *Intelligent scissors for image composition*, In: Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques—SIGGRAPH '95 (S. G. Mair et al., eds.), Los Angeles, CA, 1995, ACM, New York, NY, USA, 1995, pp. 191–198.

[14] RAMANATH, R.—KUEHNI, R. G.—SNYDER, W. E.—HINKS, D.: *Spectral spaces and color spaces*, Color Res. Appl. **29** (2004), 29–37.

[15] RODRÍGUEZ, J. T.—GUADA, C.—GÓMEZ, D.—YÁÑEZ, J.—MONTERO, J.: *A methodology for hierarchical image segmentation evaluation*, In: Proc. of the 16th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems—IPMU '16, Eindhoven, The Netherlands, 2016 (J. P. Carvalho et al., eds.), *Commun. Comput. Inf. Sci.,* Vol. 610, Springer-Verlag, Berlin, 2016, pp. 635–647.

[16] SHI, J.—MALIK, J.: *Normalized cuts and image segmentation*, IEEE Trans. Pattern Anal. Mach. Intell. **22** (2000), 888–905.

[17] SOBEL, I.: *An isotropic 3x3 image gradient operator*. In: *Machine Vision for Three Dimensional Scenes* (H. Freeman, ed.), Acad. Press, Inc. Orlando, FL, USA, 1990, pp. 376–379.

[18] SONKA, M.—HLAVAC, V.—BOYLE, R.: *Image Processing, Analysis and Machine Vision*. Chapman and Hall Computing, Gran Bretaña, UK, 1993.

[19] SURAL, S.—QIAN, G.—PRAMANIK, S.: *Segmentation and histogram generation using the hsv color space for image retrieval*. In: IEEE Proceedings in International Conference on Image Processing 2:II–II, Rochester, NY, USA, 2002, IEEE, pp. II–589.

[20] TETALI, P.: *Random walks and the effective resistance of networks*, J. Theoret. Probab. **4** (1991), 101–109.

[21] TKALCIC, M.—TASIC, J.: *Colour spaces: Perceptual, historical and applicational background*, In: Proc. of the IEEE Region Conference EUROCON '03. Computer as a Tool, Ljubljana, Slovenia, 2003, pp. 304–308.

[22] TRÉMEAU, A.—COLANTONI, P.: *Regions adjacency graph applied to color image segmentation*, IEEE Trans. Image Process. **9** (2000), 735–744.

Received December 13, 2017

C. Guada
J. T. Rodríguez
Facultad de Ciencias Matemáticas
Universidad Complutense de Madrid
Plaza de las Ciencias, 3
E–28040 Madrid
SPAIN
E-mail: cguada@ucm.es
jtrodrig@mat.ucm.es

D. Gómez
Facultad de Estudios Estadísticos
Universidad Complutense de Madrid
Av. Puerta de Hierro s/n
E–28040 Madrid
SPAIN
E-mail: dagomez@estad.ucm.es

J. Montero
Facultad de Ciencias Matemáticas
and
Instituto IGEO (CSIC-UCM)
Universidad Complutense de Madrid
Plaza de las Ciencias, 3
E–28040 Madrid
SPAIN
E-mail: monty@mat.ucm.es