

# Color Visualization of 2D Segmentations

Christoph Dalitz, Tobias Bolten and Oliver Christen

*Institute for Pattern Recognition, Niederrhein University of Applied Sciences, Reinarzstr. 49, Krefeld, Germany*  
*christoph.dalitz@hsnr.de, t.bolten@gmail.com, oliver.christen@ish.de*

**Keywords:** Visualization of Spatial Data, Area Voronoi Diagram, Graph Coloring, Color Metrics

**Abstract:** This paper deals with the problem of coloring a two-dimensional segmentation result in such a way that each segment has a different color with the constraint that adjacent segments have sufficiently distinct colors to be easily distinguishable by a human viewer. Our solution for this problem is based on a balanced coloring of the neighborhood graph built from the area Voronoi diagram. The balanced coloring with only a limited number of colors is subsequently modified to assign each segment a unique color. For picking the initial palette of evenly distributed colors, we propose a method that is based on the physical analogy of energy minimization in a Wigner crystal. To make the energy formula applicable to arbitrary color spaces and distance measures, we give generalized definitions for the center and radius of a color space with an arbitrary distance metric. The individual steps of our coloring algorithm have been evaluated on the PRIMA dataset.

## 1 Introduction

Segmentations of images are typically visualized as color images, in such a way that pixels belonging to the same segment have the same color, while pixels belonging to different segments have different colors (Saund et al., 2009) (Shafait et al., 2006). When the segmentation has to be checked or corrected by a human observer, it is however not sufficient to assign different colors to different segments, because adjacent segments are hard to distinguish when their colors are too similar. This applies in particular to document image segmentation, in which the segments representing text lines, paragraphs, words, characters, or similar are embedded in white background. Hard to distinguish neighboring segments are both a problem when looking for segmentation errors by visual inspection, as well as in the process of creating ground truth data for segmentation evaluation studies or training purposes.

Ideally, the chosen coloring would satisfy a generalized graph coloring problem:

Given a neighborhood graph of  $n$  segments,  $n$  colors  $c_1, \dots, c_n$ , and a distance metric  $d(c_i, c_j)$  on the colors, find a node coloring that maximizes the minimum distance between adjacent colors.

Some ordinary graph coloring problems are special cases of this problem: the “equitable coloring problem” (Pemmaraju et al., 2003), e.g., is the special case of only  $k$  different colors evenly distributed among the

$c_1, \dots, c_n$ , and  $d$  being the trivial metric. In the general case of a nontrivial metric, this is a difficult optimization problem that presumably cannot be solved in polynomial time.

Our goal in the present paper is therefore more modest: we only propose algorithms for selecting and assigning colors in such a way that the minimum distance between adjacent colors is “not too small”. Our approach consists of two steps: first the neighborhood graph is colored with only a small number of colors. Each of these colors is then considered as the center of a color cluster, whose colors are distributed among the nodes belonging to the center color.

## 2 The neighborhood graph

There are different definitions of segment “neighborhood”, some based on geometric distance and others based on adjacency. In our case, we are not interested in the nearest neighbors of a segment, but in *all adjacent* segments. It is thus natural to define the neighborhood graph as the dual graph of the area Voronoi diagram generated by the segments.

### 2.1 The area Voronoi diagram

For a set of two dimensional points  $P = \{p_1, \dots, p_n\}$ , the ordinary (or *point*) *Voronoi diagram* is a tessellation of the plane into disjoint cells such that the

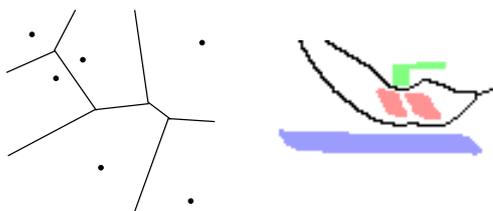


Figure 1: A point Voronoi diagram (left) and an area Voronoi diagram of three segments (right).

Voronoi cell of point  $p_i$  is given by

$$V(p_i) = \{p : \|p - p_i\| \leq \|p - p_j\| \text{ for all } j \neq i\}$$

where  $\|p - q\|$  denotes the Euclidean distance between  $p$  and  $q$ . In other words, every point in  $V(p_i)$  is closer to  $p_i$  than to each other point in  $P$ . Fig. 1 shows the resulting Voronoi cells for a sample set of six points. The *Delaunay graph* associated to a Voronoi diagram has a node for each Voronoi cell and edges between neighboring cells.

The *area Voronoi diagram* is a generalization of the ordinary Voronoi diagram in such a way that the tessellation is not generated by points, but by non overlapping objects  $S = \{s_1, \dots, s_n\}$ . Every point in the Voronoi cell  $V(s_i)$  is then closer to object  $s_i$  than to any other object:

$$V(s_i) = \{p : \min_{q \in s_i} \|p - q\| \leq \min_{q \in s_j} \|p - q\| \text{ for all } j \neq i\}$$

The right figure in Fig. 1 shows the area Voronoi cell boundaries for an example of three objects. In our application case, the objects are the segments of the segmentation. The sought neighborhood graph then corresponds to the area Voronoi diagram. Let us call this neighborhood graph the *area Delaunay graph*.

## 2.2 Computing the area Delaunay graph

As the area Voronoi diagram can be useful for bottom-up page segmentation, a number of methods have been suggested for its computation (Arcelli and di Baja, 1986) (Lu and Tan, 2005) (K. Kise, 1998). For instance, it can be computed *exactly* with a region growing approach, which can either be done directly on the image with pixel labeling, or on some representation of the segment contours. To reduce the runtime, Kise et al. suggested to compute the area Voronoi diagram *approximately* by first computing the point Voronoi diagram from contour sample points and then merging cells belonging to the same segment (K. Kise, 1998).

When only the area Delaunay graph is of interest and the exact shape of the Voronoi cells does not matter, the contour sample points can be used

to directly compute a point Delaunay triangulation. The area Delaunay graph is then obtained by merging nodes belonging to the same segment. We have chosen this approach because it can be done efficiently in  $O(n \log n)$  time for  $n$  points (Devillers, 2002). For a typical document layout, the sample points contain several collinear points which result in degenerate triangles that incorrectly have edges between non adjacent points. It is therefore necessary to clean up these triangles after computing the triangulation.

Concerning the sampling rate for the contour points, the experiments described in Sec. 4.1 have shown that taking between 20% and 30% of the contour points yields a reasonable compromise between accuracy and speed.

## 3 Coloring the graph

Our coloring algorithm for the neighborhood graph is a two step approach: first the graph is colored with a small number  $k$  of colors; then this coloring is modified so that every node obtains a different color. In two subsections, we propose algorithms for the following problems: (a) how to choose the initial  $k$  colors and the final full set of  $n$  colors, and (b) how to do the initial  $k$ -coloring and modify it to an  $n$ -coloring.

### 3.1 Color selection

The problem of color palette selection is usually considered under the optimization criterion of a best match between the palette and given images. An interesting exception is Mojsilović and Soljanin's scheme for picking a color palette from a spiral lattice in the CIE L\*a\*b\* color space (Mojsilovic and Soljanin, 2001). The resulting colors however depend on a number of input parameters and it is not even clear which input parameters lead to a given number  $n$  of valid colors in RGB space, nor have they proposed criteria for "good" parameter choices.

In our situation, we seek  $k$  colors  $\mathbf{c}_1, \dots, \mathbf{c}_k$  that are "most different" from each other. Each color  $\mathbf{c}_i$  is a vector of three values representing red, green and blue. Defining a color distance measure that models visual perception is a very difficult problem, and a wide variety of distance measures  $d(\mathbf{c}_i, \mathbf{c}_j)$  have been proposed. The best known are the Euclidean distance in RGB space  $d_{RGB}$  or the CIE L\*a\*b\* color distance  $d_{Lab}$ :

$$d_{RGB}(\mathbf{c}_i, \mathbf{c}_j) = \|\mathbf{c}_i - \mathbf{c}_j\|$$

$$d_{Lab}(\mathbf{c}_i, \mathbf{c}_j) = [(L(\mathbf{c}_i) - L(\mathbf{c}_j))^2 + (a(\mathbf{c}_i) - a(\mathbf{c}_j))^2 + (b(\mathbf{c}_i) - b(\mathbf{c}_j))^2]^{1/2} \quad (1)$$

More sophisticated distance measures include the CIE94 and the HCL color distance (Sarifuiddin and Missaoui, 2005), or the DIN99 distance (Deutsches Institut für Normung, 2001).

Concerning the “optimality” criterion for the color palette, let us first observe that in an Euclidean space, an optimal equidistant distribution is taken on by electrons repelling each other, a configuration also known as *Wigner* or *Coulomb crystal* (Rafac et al., 1991). The equilibrium distribution of these minimizes the electric potential energy

$$\varphi(\mathbf{c}_1, \dots, \mathbf{c}_k) = \sum_{i>j} \frac{e^2}{\|\mathbf{c}_i - \mathbf{c}_j\|} + \sum_i \gamma \|\mathbf{c}_i\|^2 \quad (2)$$

where  $e$  denotes the electric charge of the electrons and  $\gamma$  is the strength of a harmonic attractor in the center. It is thus natural to define as a criterion for the optimal color palette that it minimizes (2) with the Euclidean distance  $\|\mathbf{c}_i - \mathbf{c}_j\|$  replaced with a color distance measure  $d(\mathbf{c}_i, \mathbf{c}_j)$ . As we cannot use the color white  $\mathbf{w}$  (reserved for the image background) and additionally would like to keep the color black  $\mathbf{b}$  for noise left over from an incomplete segmentation, we add the “potential energy” stemming from these two points to (2). This means that we seek to minimize

$$\varphi(\mathbf{c}_1, \dots, \mathbf{c}_k) = \sum_{i>j} \frac{1}{d(\mathbf{c}_i, \mathbf{c}_j)} + \sum_i \gamma \left( d(\mathbf{c}_i, \boldsymbol{\mu}) \right)^2 + \sum_i \frac{\alpha}{d(\mathbf{c}_i, \mathbf{w})} + \sum_i \frac{\beta}{d(\mathbf{c}_i, \mathbf{b})} \quad (3)$$

where each color  $\mathbf{c} = (c_r, c_g, c_b)$  is subject to the constraint  $0 \leq c_r, c_g, c_b \leq 255$ , and  $\boldsymbol{\mu}$  is the center of the color space (see Sec. 4.2 for a precise definition of this center). The constants  $\alpha$  and  $\beta$  have been introduced for convenience so that it is possible to optionally enforce a smaller or larger distance of the colors from black or white.

In Sec. 4.2, the resulting color distribution for different color metrics and  $k = 6$  (the minimum number of colors for the coloring algorithm described in Sec. 3.2) is given. Interestingly, the best results are not obtained with the more sophisticated distance measures, but with the simple Euclidean distance in the RGB space.

For the final full set of  $n$  different colors, we consider each of the initial  $k$  colors  $\mathbf{c}_1, \dots, \mathbf{c}_k$  to be the core of a “color cluster” of  $n/k$  colors. We subsequently add colors one-by-one to a cluster around  $\mathbf{c}_i$ , such that each added color fulfils two conditions:

- on the discrete RGB color grid, it is the neighbor to one of the colors that are already in the clusters
- among all possible new neighbors, the neighbor  $\mathbf{c}$  with the closest distance to the color center  $\mathbf{c}_i$  is chosen.

### 3.2 Balanced coloring and its modification

As the area Delaunay graph is planar, it can be colored in linear time, e.g. with the 6-COLOR algorithm by Matula et al. (Matula et al., 1980). This algorithm, however, does not produce a balanced coloring, because among the colors not yet used in a node neighborhood it always picks the “smallest” color. We have replaced this color selection step with a simple heuristic: instead of the smallest color, we select the color that has been used least frequently so far. This simple rule lead to an equitable coloring in almost all of our experiments described in Sec. 4.3.

To modify the initial balanced  $k$ -coloring into a complete  $n$ -coloring, the  $n/k$  colors of each color cluster need to be distributed among the nodes assigned to the respective color cluster. To this end, we have implemented three different methods, which all traverse the graph in a breadth-first search. They differ in the way a color for a node belonging to color cluster  $C_i$  is chosen, based on the colors already assigned to its neighbors:

- random* method: select a random yet unused color from  $C_i$
- maxdist* method: select the yet unused color from  $C_i$  with the greatest distance to the neighbor node colors
- coredist* method: select the yet unused color from  $C_i$  with a distance to its closest neighbor node color similar to the distance between the color cluster cores

In our experiments (see Sec. 4.4), the random and the coredist method were comparable and better slightly worse <sup>1</sup> than the maxdist method. For most practical purposes, the simple random method will thus be a sufficient solution, that is both easy to implement and efficient.

## 4 Experiments and results

We did a number of experiments to evaluate the individual steps of our algorithm, which we have implemented in C++ within the Gamera framework for document analysis and recognition <sup>2</sup>. As document test images, we used the PRIMA dataset (Antonacopoulos et al., 2009) which, at the time of our experiments, consisted of the 55 test images from the page segmentation competition of the two ICDAR conferences in 2007 and 2009. As the segments to be colored, we

<sup>1</sup>typo in original paper

<sup>2</sup><http://gamera.sf.net/>

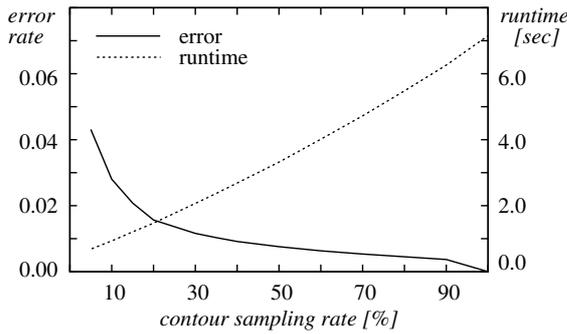


Figure 2: Effect of the contour sampling rate on the edge error rate and on the runtime.

did not consider the segments marked in the ground truth data, but the connected black components, because these were much more segments (about 5200 per image).

#### 4.1 Sampling rate for the Delaunay graph

For different contour sampling rates, we have measured the number of edges in the resulting Delaunay graph that differ from a reference graph with a 100% sampling rate. The average results over all images, with the runtimes measured on an Intel Core2 1.66GHz processor, are shown in Fig. 2. It can be seen that a sampling rate between 20% and 30% yields an error rate already below 1.5%, thereby setting a reasonable compromise between accuracy and speed.

#### 4.2 Optimal initial $k$ colors

To numerically compute the color configuration that minimizes Eq. (3), we have used the Monte Carlo algorithm described by Rafac et al. in (Rafac et al., 1991). It simply consists in moving downhill from  $N$  randomly chosen starting points and picking the global minimum from these local minima. We chose  $N = 1000$ . When applying Eq. (3), the question arises

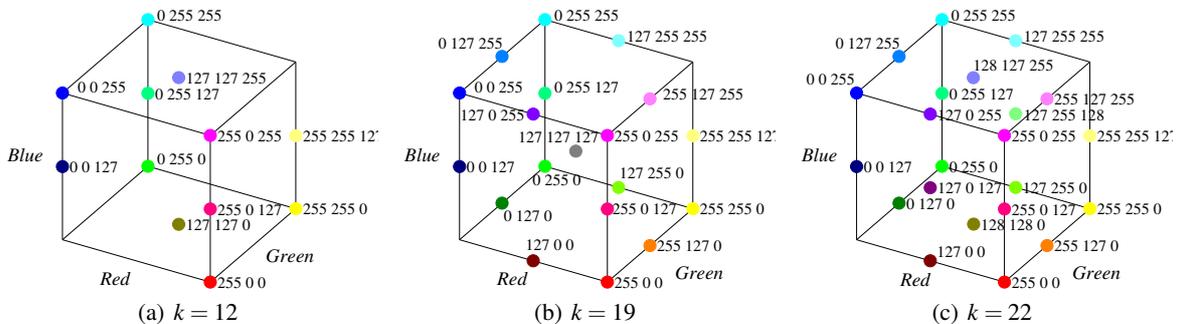


Figure 3: Minimum energy configurations for  $k$  colors in the Euclidean RGB space.

	$\mu$	$r$
RGB	(127, 127, 127)	220
L*a*b*	(135, 130, 126)	142
HCL	(210, 209, 209)	311
CIE94	(133, 137, 107)	83
DIN99	(128, 124, 120)	82
HSV	(194, 194, 194)	1.25

Table 1: RGB values of center and radius of different color space according to Eqs. (5) and (6).

how to choose the attracting strength  $\gamma$ , which is related to the radius  $r$  of the Coulomb cluster. For an unconstrained Euclidean space, (Rafac et al., 1991) gives the relation

$$2r \approx k^{0.4687} (2\gamma)^{-1/3} \quad (4)$$

Solving for  $\gamma$  and setting  $2r = \sqrt{3} \cdot 255$  yields the relation how to increase  $\gamma$  with  $k$  in the Euclidean RGB space. The resulting minimum energy configurations for  $\alpha = \beta = 1$  for different values of  $k$  are displayed in Fig. 3.

Generalizing Eqs. (3) and (4) to an arbitrary color space first requires an appropriate definition of the center color  $\mu$  and the radius  $r$  of the color space. A generalized definition of the arithmetic mean of  $x_1, \dots, x_N$  can be derived from its property to minimize  $\sum_i \|x_i - \mu\|^2$ . We thus define the center color  $\mu$  and the radius  $r$  of a color space with color distance  $d$  as

$$\mu = \arg \min_{\mathbf{c}} \sum_{\mathbf{c}} d(\mathbf{c}, \mathbf{m}) \quad (5)$$

$$r = \max_{\mathbf{c}} d(\mathbf{c}, \mu) \quad (6)$$

where  $\mathbf{c}$  runs over all possible RGB colors. The numerical values resulting from this definition are listed in Tbl. 1 for six different distance metrics: the Euclidean distance in RGB and L\*a\*b\* space, the DIN99 distance (Deutsches Institut für Normung, 2001), the CIE94 and HCL distance (Sariffuddin and Missaoui, 2005), and the cylindrical distance in HSV space (Ikonomakis and Plataniotis, 1999).

Using the values from Tbl. 1 and with  $\alpha = \beta = 1$ , we have determined the minimum energy choice of

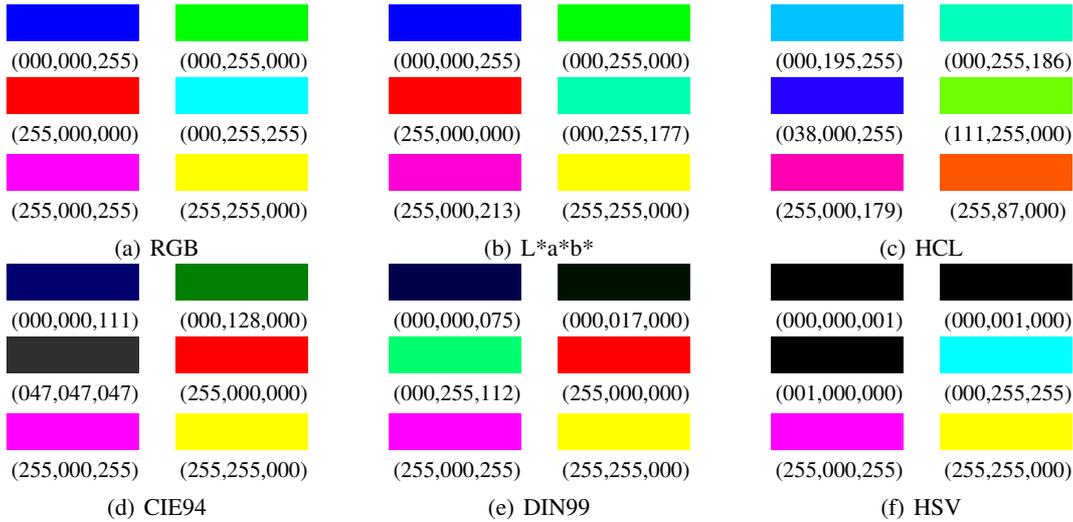


Figure 4: The color distributions minimizing Eq. (3) with respect to different color distance metrics.

$k = 6$  colors. This is a natural choice for  $k$ , because it is the minimum number of colors required by the graph coloring algorithm 6-COLOR. Moreover, in this case the energy minimum with respect to the Euclidean distance in RGB is already known (six colors in the corners of the RGB cube), which provides a nice benchmark against which we can compare other color distance metrics. Fig. 4 shows the results for the six different distance metrics.

Both the HCL and L\*a\*b\* distance result in a poorer distribution than the RGB distance, because both include two similar green hues. Interestingly, the same HCL and L\*a\*b\* results are obtained when the RGB result is used as a starting point for the minimization algorithm: both distance metrics make the distribution worse. Even though the CIE94 distance results in six visually different colors, one of them is almost black, despite the repelling potential from the  $\beta$ -term in Eq. (3). The same applies to the DIN99 distance which even leads to two colors that are almost black. For the cylindrical HSV distance, we obtain even three almost black colors, but this is due to a fundamental shortcoming of this distance measure in case of colors that have a low intensity, but a high saturation because one of the primary colors is zero.

Our experiments thus provide a justification for the intuitive approach to select colors in the corners of the RGB cube, even though this includes the somewhat similar colors green (0,255,0) and cyan (0,255,255). Finding a color distance measure and an optimization criterion that perceptually improves this distribution is an interesting area of further research.

### 4.3 Balance of the initial $k$ -coloring

A coloring is *equitable*, when the color counts do not differ by more than one (Pemmaraju et al., 2003). Our balanced variant of the 6-COLOR algorithm produced indeed equitable colorings for 51 of the 55 test images. In each of the remaining four images, there was one color count difference of two and the other counts differed not more than one. This shows that the simple heuristic color selection rule indeed results in a balanced coloring.

### 4.4 Quality of the final $n$ -coloring

To compare the quality of the three color distribution methods described in Sec. 3.2, we have measured both the minimum and the average RGB distance between neighboring node colors. Tbl. 2 lists the averaged results over the  $N = 55$  test images. Both the minimum and average distances decrease with the number  $k$  of initial colors. This shows that our approach of building color clusters around few initial colors is an improvement over the naive approach of picking  $n$  evenly distributed initial colors, which would be the extreme case  $k = n$ .

All three methods for modifying the  $k$ -coloring into an  $n$ -coloring lead to similar mean and minimum color distances between neighboring nodes. The simple maxdist method, which always picks the most far away color among all free colors from a color cluster, leads to the largest distance. Even though the random method leads to smaller distances, its runtime is only linear in the number of nodes, while the runtimes other two methods are quadratic in the number of nodes. The random assignment of colors from a

	<i>method</i>	<i>mean dist</i>	<i>min dist</i>
$k = 6$	<i>maxdist</i>	325.6	239.3
	<i>random</i>	325.1	234.6
	<i>coredist</i>	325.3	235.4
$k = 16$	<i>maxdist</i>	259.8	120.0
	<i>random</i>	259.0	115.0
	<i>coredist</i>	259.6	115.4
$k = 26$	<i>maxdist</i>	233.8	77.4
	<i>random</i>	233.3	73.1
	<i>coredist</i>	233.6	73.9

Table 2: Mean and minimum RGB color distance between neighboring nodes for the three color distribution methods and different numbers  $k$  of core colors, averaged over all images from the PRIMA dataset.

cluster to its nodes will therefore often be a sufficient and reasonable approach in practice.

## 5 Conclusions

The methods proposed in this paper provide a practical solution for using colors as segmentation labels in such a way that the segmentation is easily visible in the resulting color image for a human observer.

Nevertheless, there remain two interesting areas of further research. One consists in finding better ways for selecting the initial  $k$  colors, both with respect to the optimization criterion Eq. (3) and to an appropriate color distance measure. As the latter problem of defining a perceptual color distance is a very difficult problem still waiting for satisfactory solutions, it might alternatively be more promising to perform psychological experiments for directly selecting the  $k$  perceptually “most different” colors. The other open problem is the more fundamental algorithmic question whether there is an efficient exact or approximate solution for the generalized graph coloring problem stated in the introduction.

## REFERENCES

- Antonacopoulos, A., Bridson, D., Papadopoulos, C., and Pletschacher, S. (2009). A realistic dataset for performance evaluation of document layout analysis. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 296–300.
- Arcelli, C. and di Baja, G. S. (1986). Computing voronoi diagrams in digital pictures. *Pattern Recognition Letters*, 4:383–389.
- Deutsches Institut für Normung (2001). Farbmétrische Bestimmung von Farbabständen bei Körperfarben nach der DIN99-Formel. *DIN 6176*.
- Devillers, O. (2002). The delaunay hierarchy. *International Journal of Foundations of Computer Science*, 13:163–180.
- Ikonomakis, N. and Plataniotis, K. (1999). A region-based color image segmentation scheme. In *Visual Communications and Image Processing (VCIP)*, pages 1201–1209.
- K. Kise, A. Sato, M. I. (1998). Segmentation of page images using the area voronoi diagram. *Computer Vision and Image Understanding*, 70:370–382.
- Lu, Y. and Tan, C. (2005). Constructing area voronoi diagram in document images. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 342–346.
- Matula, D., Shiloach, Y., and Tarjan, R. (1980). Two linear-time algorithms for five-coloring a planar graph. *Technical Report STAN-CS-80-830, Stanford University*.
- Mojsilovic, A. and Soljanin, E. (2001). Color quantization and processing by fibonacci lattices. *IEEE Transactions on Image Processing*, 10:1712–1725.
- Pemmaraju, S., Nakprasit, K., and Kostochka, A. (2003). Equitable colorings with constant number of colors. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 458–459.
- Rafac, R., Schiffer, J., Hangst, J., Dubin, D., and Wales, D. (1991). Stable configurations of confined cold ionic systems. *Proc. Ntl. Acad. Sci. USA*, 88:483–486.
- Sarifuddin, M. and Missaoui, R. (2005). A new perceptually uniform color space with associated color similarity measure for content-based image and video retrieval. In *ACM SIGIR Workshop on Multimedia Information Retrieval*, pages 1–8.
- Saund, E., Ling, J., and Sarkar, P. (2009). Pixlabeler: User interface for pixel-level labeling of elements in document images. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 646–650.
- Shafait, F., Keysers, D., and Breuel, T. (2006). Pixel-accurate representation and evaluation of page segmentation in document images. In *International Conference on Pattern Recognition (ICPR)*, pages 872–875.