Optimal palette extraction as part of scanned graphics vectorization

Vinciane Lacroix and Mahamadou Idrissa; Royal Military Academy; Brussels/Belgium

Abstract

*The proposed scanned graphics palette extraction process starts with the extraction of colors of uniform regions, then the colors of local features. In order to enhance the discrimination of unsaturated colors, the image is converted into L*a*b* space and the CMC color difference is used together with the Euclidean distance. After the palette extraction, a regularization process modifies the color of a pixel in order to make it coherent with its neighborhood, thus reducing the number of connected components, hence providing one step further to vectorization. Results are shown on scanned maps and on some scanned graphics.*

Introduction

Vectorization is the process which converts a bitmap image, i.e. a multi-matrix type of structure, into a set of vectors associated with specific attributes (color, width, etc.). Vectorized data offer many advantages: they may be edited, they capture the relevant information in an abstract form and thus offer a high compression rate. The process is thus useful for image compression, image understanding, and content-based image retrieval.

The problem is not well-posed, as different vector representations may generate the same raster image. Vectorization thus imply getting the "best" vector representation producing the considered bitmap.

According to a recent study [6], approximately 13.5 million images are vectorized in the United States every years, consuming more than 7 million man hours. These images are made of photos, artworks, logos, etc. Despite this demand, there has been limited research done on colored image vectorization except from a specific application: scanned maps [3], [5], [16] for which the vectorization is performed on each colored layer, thus after the color extraction process. Vectors may then for example be introduced in Geographical Information Systems to produce editable maps, or projected on top of some Remote Sensing data.

Commercial vectorization software [1] exist but do not provide satisfying results in a full automated mode [9]. In his Phd Thesis, Diebel [6] compares his "Vector Magic" (http://vectormagic.com/home) vectorization tool with professional commercial packages proposed by the most important players in this field: Adobe, Microsoft and Corel, and demonstrates the superiority of Vector Magic, while recognizing the good quality of Adobe and Corel software packages, despite the fact that the latter require fine tuning of some parameters while "Vector Magic" may work in a full automated mode.

In the following, we start with the capabilities of an ideal raster-to-vector (R2V) tool. As a first step to reach this goal, we propose to compute an intermediate simplified image. First, the optimum palette is extracted. At this stage all pixels are assigned a label corresponding to the nearest color. Then a regularization step updates the label of a pixel in order to make it coherent with its neighborhood. The results of the optimum palette extraction method is compared with available software and the results of the regularization process are shown for several maps and scanned graphics.

An ideal raster to vector tool

The ideal R2V tool should be able to extract points, lines and regions from the image and provide for them the characteristics that a basic vector graphics editor offers.

Points should have a color attribute which may be defined as a triplet. Should a point have a size attribute? Probably not, otherwise it would be better described as a region. A point should then be completely defined by its location in the plane: $p(x,y)$, and its color $c(c_r, c_g, c_b)$ where x and y are continuous coordinates and c_r, c_g, c_b are the respective components in Red, Green, Blue, if the RGB color space is used.

Lines should also have a color attribute, but probably not a width for the same reason. They may be represented as lists of points p_i, linked by linear, circular or elliptic segments or as piecewise cubic representations such as Bezier curves for example. Lines may further be characterized by a "style" such as continuous, dashed, or dotted. The best R2V tool will recognize the style of the line, while an acceptable one will identify separately each element of the line.

Finally regions should be characterized by a border described as a line without color nor style. Regular shapes such as rectangles, circles and ellipses should have a geometric description; elongated regions should be described by their median axis and a local width along this axis. A region could be uniform, gradually filled, or textured. A uniform region requires a constant color attribute, gradient filled region requires two colors and a rule for the evolution of the fill, and finally textured region requires the generation rule of the texture. Indeed, in a textured area, what matters is not the exact position of the micro-features present in the texture, but rather the shape and local distribution of the latter.

In this framework, thin characters should be described as a set of lines and thick ones as regions. Their recognition should be left to an Optical Character Recognition tool (OCR).

The problem of textured region vectorization is illustrated in Figure 1. A textured region generated by a graphical tool is shown at the left. A scanned version in which the pixel size is similar to the micro-feature will produce anti-aliased pixels, as the probability of the micro-feature to be located exactly on the same grid is small. A good R2V tool would perform a "vectorial" decomposition of such a textured region providing its set of micro-features on a uniform background, while the best one would provide a more abstract definition of the texture.

Today, such an ideal R2V tool performing these tasks as a whole on color images belongs to science fiction, although much research have been done on specific separated tasks such as lines or circles extraction (see [17] and [9]) mainly on binary images.

As a more realistic but still challenging objective, we require from our future R2V tool to be able to generate a description of the image as a set of points, lines and regions of uniform

[1] //en.wikipedia.org/wiki/Comparison_of_raster_to_vector_conversion_software

©2010 Society for Imaging Science and Technology

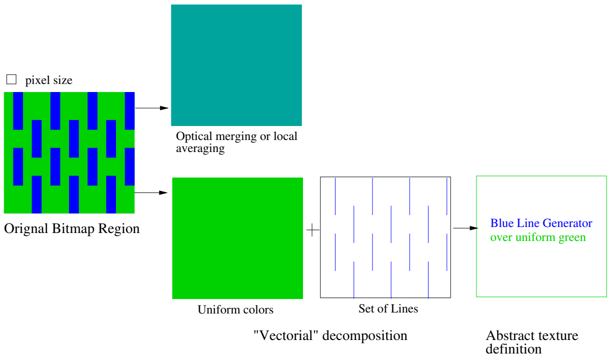

Figure 1. *Vectorization of a textured region*

color, and if a texture is present in some region, each of its micro-elements will be identified separately.

Overview of the proposed method

The envisaged R2V tool may be performed by a three-phases processing.

In this article we will make use of the following notations:

- Let (i, j) denotes the pixel p located at row j and column i of a bitmap image,
- let $\mathscr{P} = \{c_1, \ldots, c_i, \ldots, c_n\}$ denotes the set of colors that will be assigned to the pixels
- let $\mathscr{L} = \{\lambda_1, \ldots, \lambda_i, \ldots, \lambda_n\}$ be the labels (or index) associated with these colors
- let λ_p denotes the label associated with the pixel p,
- let I^λ, I, and I^c denote the labeled image, the original bitmap color image and the image in which the color at p has been replaced by the one associated with its label λ_p respectively.

The aim of the first phase is to identify the best color palette $\mathscr{P}$ characterizing the image. This task is called "optimum palette extraction".

The aim of the second phase is to find the best labeled image that gave rise to the original bitmap, that is to find I^λ from the original image I. Thus, at each pixel, we shall find the "maximum a posteriori estimate" (MAP) $\hat{\lambda}$ using Bayes rule: $\hat{\lambda}_p = \lambda$ for $P[\lambda_p = \lambda | I] = max_{\lambda_i \in \mathscr{L}} P[\lambda = \lambda_i | I]$.

For this phase, we use a Markov Random Field (MRF) model [7] which assumes that the probability of a label λ, given all other labels is equal to the probability of the label λ, given the labels of its neighborhood only. We start with the labeling corresponding to the assignment of each pixel to the nearest color and update the labeling so as to minimize an energy function defined as the sum of two terms. The first term penalizes the differences between the original image I and the labeled image I^λ while the second penalize improbable local configurations. Such models have been widely used in image segmentation [2].

Finally the aim of the last phase is to transform the connected components into vectors.

In this contribution, we will address the first two phases of the suggested vectorization process.

Optimum palette extraction

Optimum palette extraction can be seen as a specific clustering problem. While many articles can be found on palette extraction of photographs, only a few of them deal with graphics. The latter require in general a much smaller palette and could

[2]www.visionbib.com/bibliography/segment369.html

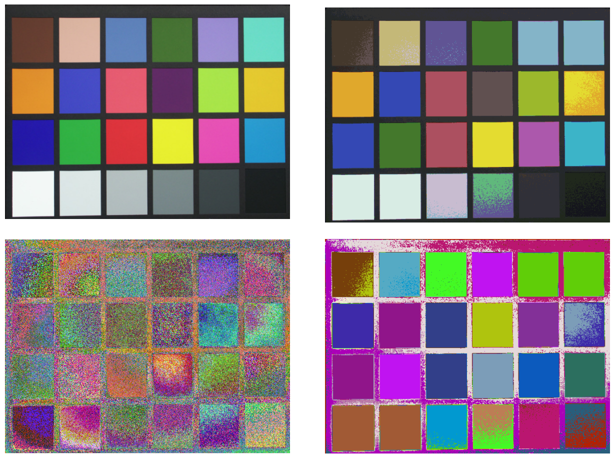

Figure 2. *Palette extraction: (left) photograph of a Macbeth color Chart (right) results of median-cut (tiffmedian with* $n = 28$*)(up: true colors, down: false colors)*

therefore benefit from higher compression rate, and a representation closer to their vectorial representation. Moreover, graphics are usually *conceived* with a limited number of colors. When the number of colors in the palette is a power of two, the process is also known as "color image quantization". This field has been widely studied during the eighties and nineties. For a comparison of color image quantization methods see [2]. One popular method is the median-cut algorithm [8] even proposed as the linux command "tiffmedian"; it requires n, the number of colors of the reduced palette (where n is not limited to powers of 2). The method is fast but the results are not of good quality as can be seen on Figure 2. The image taken from [14] is a photograph of a Macbeth color chart. In order to judge both the difficulty of the image and the quality of the results, the images are also displayed in pseudo colors.

Evaluation of palette extraction requires both qualitative and quantitative tests. The qualitative tests involve the visual comparison of the original image with the reduced-colored image. Quantitative tests rely on the computation of a distance between the original and the reduced-colored image.

Apart from the noise present in the image, the main difficulty of automatic palette extraction of scanned graphics comes from mixed pixels resulting either from the superposition of ink in the printing process or from anti-aliasing. Anti-aliasing occurs at the border of regions of different colors. When the region is a line, the proportion of such mixed pixels might be as high as one third: the pixel lying exactly on the medial axis of the line might get a color close to the one expected, while at both sides, pixels will be made of a mixture of the line color and the neighboring region color. If the scanning resolution is too coarse, the width of thin lines may be smaller than the pixel size, so that anti-aliasing may also occur on pixels located on the medial axis.

Given this analysis, as far as the palette extraction is concerned, we propose in this contribution a strategy for selecting the data candidates in order to ignore pixels where anti-aliasing occurs and the introduction of the CMC color difference [13] not only for a better discriminative power, but also for quality evaluation.

Selecting data candidates

As the palette should represent the most representative *perceived* colors, the image is transformed into the *uniform* L*a*b* space (noted "Lab" in this paper), in which the Euclidean distance reflects the perceived distance. Note that this is actually not true for low-saturated colors.

Graphics information is contained in points, lines and regions, but not at their interface made of anti-aliased pixels. Edges should thus be removed from the data candidates. Statistically, as about 90% of the image edges [10] are present in the intensity image, ignoring edge pixels resulting from a non-maximum suppression of the gradient norm of the Luminance (a Gaussian gradient with $\sigma = 0.7$ is recommended) will remove most of the anti-aliased pixels.

Given that:

- the color of thin features may be affected by anti-aliasing,
- the color of points and lines will most probably also be present in uniform regions,
- uniform regions are perceptively more important,

we suggest to use the median-shift algorithm [12] to extract the colors of uniform areas first, then to extract the not yet discovered colors of local features.

Thresholding the Gaussian gradient norm of the Luminance provides the pixels lying in uniform regions (a threshold of 1.8 is recommended).

The median shift

The median-shift [12]is an iterative process which shifts each data point to the "median" point of its neighborhood defined as all the data points located at a distance lower or equal to R. Data points are viewed as a node in a graph, where nodes are connected if they belong to the same neighborhood. The "median point" is defined as the point which has as ith component the median of the ith components of all points in the neighborhood. The process converges to a set of "clusters" made of one or several connected nodes. The distance beween two nodes belonging to different clusters is larger than $R/2$.

Therefore, we suggest that all the pixels at a lower distance than $R/2$ are considered as correctly assigned. In order to better discriminate between unsaturated colors, in this paper, the distance considered is the maximum of the Euclidean distance and CMC distance [13]. If the percentage of non-assigned uniform pixels is higher than a threshold (a threshold of 1% of the initial data set is recommended) , the median-shift is repeated to get additionnal clusters. Indeed, during the median-shift process, some color distribution maybe shifted to a close (but nevertheless different) more dominant color distribution.

The next candidates to consider should be pixels that are not yet assigned and located at the center of local features. Considering all pixels but edge pixels give good quantitative results, but considering among them only the local maxima and minima of the Luminance and the maxima of the saturation (expressed as $\sqrt{(}a^2+b^2)$ provide better quantitative results and speed up the process. The rationale of this choice lies on the fact that anti-aliased colors are less saturated and either less luminous or less dark than the palette colors.

In color reproduction industry a local color difference below 8 is recommended [4]. A radius range of [16-19] thus seems appropriate and a value of $R = 18$ will in general provide good results. In our experiments we have chosen in the range [16-19] the radius such that $n * E_{data}(I^{\lambda}, I)$ is the minimum, where n is the number of colors in the palette and $E_{data}(I^{\lambda}, I)$ is defined in Equation 1.

Implementation

A typical map is about 64 cm by 40 cm. Recommended scan resolution varies between 300 to 600 samples per inch, so that a scanned map can be as large as 10078 pixels by 6299 pixels. A 512 by 512 sample could be too small to have the chance to get all pixel colors, while four of them will most probably do. We therefore recommand to extract four random cuts of 512x512 and on each of them perform the following process:

```
convert to L*a*b*
compute gaussian gradient of L*
build set A= pixels with norm<1.8
      (mainly pixels in uniform areas)
build set E= non-max suppression of norm
extract palette1 using median-shift on A
perform line extraction on L*
  build set F=  non-max suppression of line norm
      (mainly pixels located on features)
  build set SF=subset F
            = local max of L* and saturation
              not present in E (optionnal)
      (less subject to anti-aliaising)
classify pixels of A and SF with distance <R/2
  build set U= unclassified pixels
extract palette2 using median-shift on U
merge the two palettes= palette_cut
```

Once the palette of each cut is obtained, each color of the palette gets a weight corresponding to the number of non-edge pixels assigned to it. A new median-shift is then made on the whole set of colors.

Regularization

We start with the labelling corresponding to the assignement of each pixel to the nearest palette color and update the labelling so as to minimize an energy function $E(I^{\lambda})$ defined as the sum of two terms.

$$E(I^{\lambda}) = E_{data}(I^{\lambda}, I) + \beta E_{constraint}(I^{\lambda})$$

The first term called data term expresses the cost of assigning a color label λ to a pixel p.

$$E_{data}(I^{\lambda}, I) = \sum_{p} E_1(\lambda_p) \qquad (1)$$

where E_1 expresses the cost of assigning to the pixel p the color of λ_p in the color palette. The most obvious choice for E_1 is to consider the distance between these two colors. As for the palette extraction phase, the considered distance is the maximum of the Euclidian and the CMC color difference.

The second term gives the cost of assigning to the pixel p the color index λ_p given the labelling of its neighborhood:

$$E_{constraint}(I^{\lambda}, I) = \sum_{p} E_2(\lambda_p, \aleph(p))$$

where E_2 expresses the cost of some specific local configuration defined in $\aleph(p)$, the neighborhood of p.

The simplest choice for E_1 is a function that impose a penality for neighbouring pixels having different color labels:

$$E_2(\lambda_p, \aleph(p)) = \sum_{q \in \aleph(p)} V(\lambda_p, \lambda_q)$$

where $V(\lambda, \lambda_q) = \{ \begin{array}{ll} 1 & \text{if } \lambda \neq \lambda_q \\ 0 & \text{otherwise} \end{array}$ and $\aleph(p)$ is the set of 4 or 8 neighbouring pixels of p.

Note that this choice of E_2 favours more regions than lines, and will avoid to keep isolated points unless E_1 is very low.

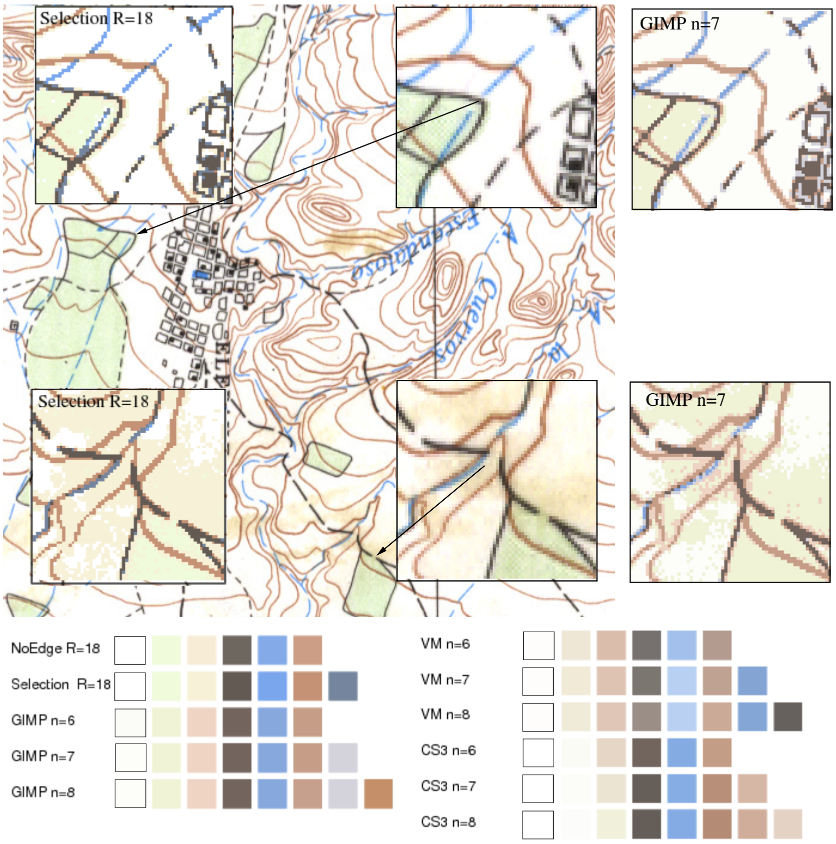

Figure 3. *Optimum palette extraction of a low saturated image. Two zooms are overlaid (middle). Results on these zooms of the proposed palettization and of the GIMP palettization are shown on the left and right respectively. Below the images: palettes extracted by all methods.*

Figure 4. *Optimum palette extraction of a saturated image. Two zooms are overlaid (middle). Results on these zooms of the proposed palettization and of the GIMP palettization are shown on the left and right respectively. Below the images: palettes extracted by all methods.*

The parameter β controls how much the constraints should be considered in the optimal solution. A good regularization depends on the appropriate choice of E_1, E_2 and β. The simplest optimization method for the energy function $E(I^\lambda)$ is an exhaustive search procedure through all possible label value $\lambda \in \mathscr{L}$ and for all the pixels p in the image I. An alternative is to use simulating annealing, a stochastic optimization procedure, which avoids to be trapped in local minima.

Constraining the edge pixels to have a label present among its non-edge neighbours only will reduce further the complexity of the resulting image.

Results

We have applied the optimum palette extraction to several maps and a photography of the Macbeth color chart [14], using the proposed strategy and compared it the Vector Magic color extractor, the GIMP and CS3 (Photoshop) palette reduction tools, through their "Image>mode>indexed" command, and assigning the number of colors. In GIMP, this number is indicative: we have observed sometimes one class more or less. CS3 offers three different method: perceptual, selective and adaptive. In all experiments we have used the "perceptual" as it was providing a better qualitative palette. Vector Magic color extractor allows a maximum of 12 colors which is not sufficient for some images. These methods have been chosen because they were performing better than other palette reduction tools (see [12]). Despite the fact that they were not designed for this specific task, they perform it very well! The two-phases median-shift is used either with a selection of non-edge pixels, or with all non-edge pixels as described above. Tests were performed on 512x512 images.

The qualitative test is based on the visual comparison of the extracted palettes and of the images after palettization. Figure 3 shows part of a map and two zooms showing the effect of aliasing (up-middle and bottom-middle). The results of the two best methods on these zooms are shown: GIMP with n=7 (right) and the proposed method with $R = 18$ (left). Notice on the bottom zoom that GIMP is unable to discriminate the green from the beige of the shadow, as a consequence, the green of the palette is more brownish than what it should be. For all the other methods, only the palettes are displayed. Note that all these methods hardly identify some green in the image.

Another test on a more saturated image can be seen on Figure 4. Again, part of the map is shown as background with two zooms overlaid in the middle of the image; results of the proposed method and of the GIMP method are overlaid at the left and the right of these zooms respectivelly. Note that the blueish green and the violet seen in the upper zoom are only detected by the two-phases median-shift. The yellow green present in the lower zoom is also detected by that algorithm while GIMP is unable to get this color in its palette. Note that the palette is quite stable when R changes from one unit, and that the palette extracted using a reduced number of pixels is quite similar with the one that considers all non-edge pixels, but is about half time faster.

The quantitative test is based on the distance between the original image and the color-reduced image as in Equation1, divided by the total number of pixels N:

$$D(I^\lambda, I) = \frac{\sum_p E_1(\lambda_p)}{N}$$

For the computation of E_1, the maximum of the Euclidean distance and the CMC color difference has been considered. The results for five different images are shown in table 1. We have chosen this distance because we consider that the Euclidian distance in the Lab space was not close enough to human perception.

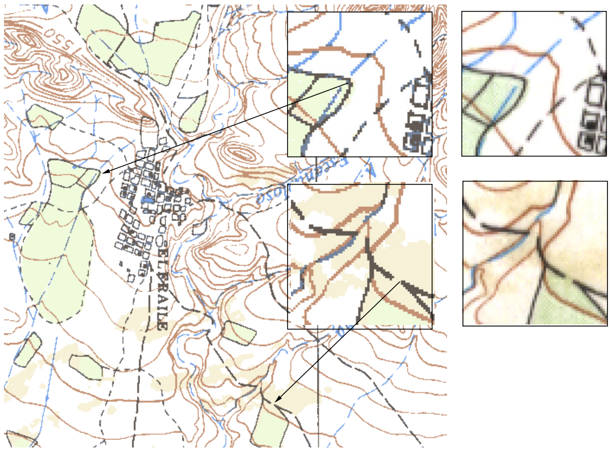

Figure 5. Regularization of the labeled low-saturated image

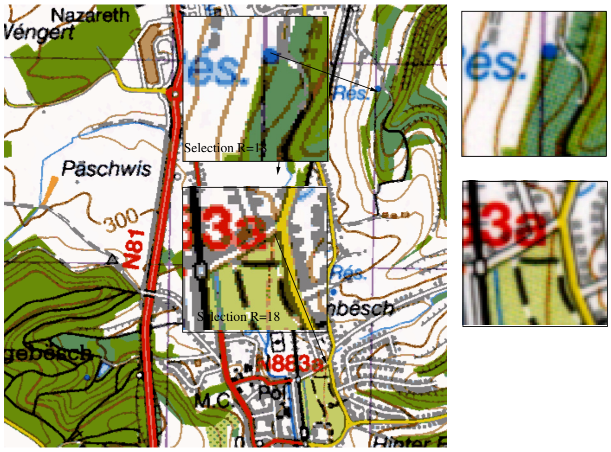

Figure 6. Regularization of the labeled saturated image

Except on the Ouzdha image, the two-phases median shift algorithm performs better than the other two algorithms with respect to this distance, but, even if the proposed distance is better than the Euclidian distance, it does still not reflect the percieved distances between the images.

images	n colors	two phases	GIMP	CS3
mapa4	7	6.9756	7.58052	7.5362
Ouzdha	11	16.0243	11.2303	15.0082
Chart	28	2.6226	2.63489	8.40815
mapa8	26	5.00196	5.40047 (27)	5.77837
Pristina	12	6.11736	7.71247	6.84294
Virton	22	9.68114	11.2541 (23)	11.0182

Table 1: Mean error for several images and algorithms

The second phase involves a regularization. This process drastically reduces the number of connected components, thus simplifying the image. The result of the full process applied to the images shown in Figure 3 and 4 is shown in Figure 5 and Figure 6. As expected this regularization process favors homogeneous regions and thus may delete some local features.

©2010 Society for Imaging Science and Technology

Summary and discussion

The optimum palette extraction using the median-shift in the Lab color space with the maximum of the Euclidean and the CMC distance in a two-step strategy gives excellent results both qualitatively and quantitatively. The quantitative tests do not reflect the perception, so that another distance measure should be proposed. We have compared the method with Photoshop CS3, GIMP palette reduction tools, and, when possible, with the Vector Magic color finder on some maps and scanned graphics. A better regularization considering the nature of the underlying pixel could simplify further the image. The pixels of similar labels must still be connected in order to identify all the vectors.

Acknowledgments

This work has been funded by the Belgian Ministry of Defense as part of the C4_12 study.

References

[1] Bouman, C. A.: Cluster: An unsupervised algorithm for modeling Gaussian mixtures `http://www.ece.purdue.edu/~bouman` (1997)

[2] Braquelaire, J. P., Brun L.: Comparison and optimization of methods of color image quantization, IEEE Transactions on Image Processing, vol. 6, nb 7, pp. 1048-1052 (1997)

[3] Chen, Y. at al: Extracting Contour Lines From Common-Conditioned Topograpic Maps. IEEE TGARS, vol. 44, nb 4, pp. 1048–1057 (2006)

[4] Delmas, J.: La gestion des couleurs pour les photographes, Eyrolles, p. 57, (2005)

[5] Deseilligny, M. P.: Lecture automatique de cartes, Phd thesis, Université René Descartes, Paris, France, (1994)

[6] Diebel, J.R.: Bayesian Image Vectorization: The Probabilistic Inversion of Vector Image Rasterization. Phd thesis, Standford University, (2008)

[7] Geman, S., Geman, D. : Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images, IEEE Trans. Pattern Analysis and Machine Intelligence 6, pp. 721-741 (1984) Intelligence 6: 721-741

[8] Heckbert, P.: Color image quantization for frame buffer display, SIGGRAPH Comput. Graph., Vol. 16, nb. 3, pp. 297–307, (1982)

[9] Hilaire, X: RANVEC and the Arc Segmentation Contest: Second Evaluation. In: Wenyin Liu and Josep Lladós (eds.) Graphics Recognition—Ten Years Review and Future Perspectives. LNCS, vol. 3926, pp. 362–368. Springer Verlag, (2006)

[10] Koschan, A., Abidi, M.: Detection and Classification of Edges in Color Images. Sig. Proc. Mag., Spec. Issue on Color Img. Proc., vol. 22, nb. 1, pp. 64–73 (2005)

[11] Lacroix, V., Acheroy, M.: Feature-Extraction Using the Constrained Gradient. ISPRS J. of Photogram. and RS, vol 53, nb. 2, pp. 85–94 (1998)

[12] Lacroix, V.: Automatic Palette Identification of Colored Graphics, In: Eight IAPR International Workshop on Graphics Recognition, Univ. of La Rochelle, France, pp. 95-100, (2009)

[13] Ohta, N., Robertson, A. R.: Colorimetry Fundamentals and Applications. John Wiley & Sons, Ltd. (2005)
Workshop Farbbildverarbeitung, 7.-8. Oktober 2004, Koblenz, Germany, Proceedings. Der Andere Verlag, Tnning 2004, ISBN 3-89959-245-X

[14] Palus, H.: On color image quantization by the k-means algorithm, In: Workshop Farbbildverarbeitung, Droege, Detlev and Paulus, Dietrich (eds) Universität Koblenz-Landau, Tönning, Der Andere Verlag (2004)

[15] Price, K.: Computer Vision Biography, 8.8.3 MRF Models for Segmentation, `http://www.visionbib.com/bibliography/segment369.html` accessed 05-09

[16] Robert, R.: Contribution à la lecture automatique de cartes, Phd thesis, Université de Rouen, Rouen, France (1997)

[17] Tombre, K.: Graphics Recognition: The Last Ten Years and the Next Ten Years, Graphics Recognition—Ten Years Review and Future Pespectives, Springer Verlag,Wenyin Liu and Josep Lladós Editors, LNCS, Vol 3926, pp. 422–426, (2006)

Author Biography

Vinciane Lacroix got a "Licence en Physique" from the ULB, Bruxelles (BELGIUM), a Master in Computer and Systems Eng. from RPI, Troy, N.Y. (USA), and a Phd from ENST, Paris (FRANCE). She has been involved in Computer Vision since 1985; she started her career at Philips Research Laboratory Belgium. She joined the Royal Military Academy Belgium in 1994 to work on various remote sensing projects. Her interests include Pattern Recognition, Image Processing, Remote Sensing for security and stability, Vision, Humanitarian Demining, and more recently colored graphics vectorization.