Raster-to-vector conversion: problems and tools towards a solution a map segmentation application

Vinciane Lacroix
Signal and Image Centre,
Royal Military Academy, 1000 Brussels, Belgium
Vinciane.Lacroix@elec.rma.ac.be

Abstract

After the analysis of the problems a raster-to-vector (R2V) software can meet, a strategy involving a pre-processing, a clustering, a labeling and finally a vectorization phase is proposed. Much emphasis is put on the clustering and the labeling phases which depend on the pixel type (edge, line, or other). In particular, it is suggested to use the median-shift on all the pixels but the edgels to extract the main colors. Results are shown on scanned maps.

1. Introduction

Automatic conversion of scanned graphics has been tackled over the last 30 years and is still considered as a hot topic [Tombre, 2006]. Commercial software[1] exist but usually require human intervention except on very simple cases [Hilaire and Tombre, 2006].

Since the nineties, a specific type of graphics deserved more attention: scanned maps. Recent works involve vectorization of contour lines [Chen et al., 2006], sometimes leading to the generation of DEM [Pouderoux, 2007]. In such an application, “expert” rules related to map generation are used to constrain the extraction of specific map features [Deseilligny, 1994], [Robert, 1997].

Research studies have usually addressed one problem at a time. In particular, focus has been made on line extraction on binary images [Chhabra and Philips, 2000], [Hilaire, 2006], and, in maps, on feature extraction on each colored layer.

In the current application, the map is considered as an example of graphic, therefore no specific map knowledge is used. The objective is to transform the input raster into a set of colored lines and regions.

Evaluating a R2V conversion software is a difficult task as there is in general no unique solution. A fair evaluation should require a common data set, and a consensus about what is the “best solution”. To the author knowledge, neither the data set nor the consensus exist. In this respect, maps offer an interesting R2V application as the corresponding GIS layers provide the best solution.

1. for a comparison see http://en.wikipedia.org/wiki/ Comparison_of_raster_to_vector_conversion_software

The aim of this paper is to propose a reflection on what should be expected from a R2V conversion software, then give a strategy and some tools towards this aim. All aspects of the conversion cannot be covered here, so that the focus will be the color layers extraction, which, today, is rarely a fully automated process.

The paper is organized as follows. In section 2 some R2V conversion problems are analyzed. Section 3 proposes a reasonable strategy with specific tools for the preprocessing, the clustering, the labeling and the vectorization phases. Results are shown in Section 4 and conclusions provided in Section 5.

2. Identified problems

We assume that the input graphics have a limited number of unknown colors, thus excluding photographs. Identifying the colors is thus the first goal, then comes the vectorization on each layer. Apart from the noise generating unexpected colors, problems for color identification and attribution tasks come from aliasing, superposition of colors and textured areas.

2.1. Aliasing effects

Aliasing comes from digitalization: a pixel lying at the edge of two regions will have a “mixed” color. Consider the bitmaps on Fig 1 (a). Should a line be extracted between R and S in (b)? If the line color is “between” the colors of the regions, as it might be due to aliasing, the most simple solution is to consider it as border. The answer is then yes for the upper example and no for the bottom one. Border of regions should thus be considered with care. Continuous approximation of lines and regions is left for a later process.

2.2. Shift and superposition effects

A shift between the printed colored layers generates “holes” and superposition effects like in Fig 1 (c) where the shift of the yellow layer is generating a white line (empty space) and a green line (superposition on blue). Should the R2V software extract these lines or identify the shift? As

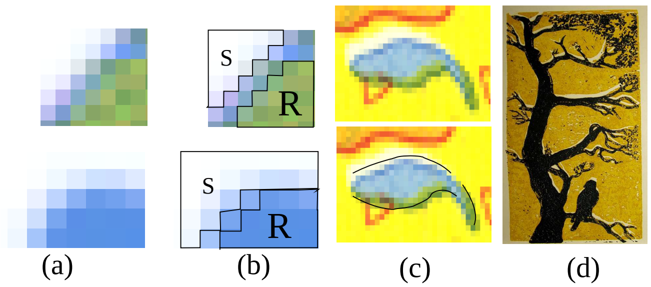

Figure 1. Bitmaps and various R2V solutions

a low level process, it should indeed identify both lines while the shift of the layer could be deduced at another stange, depending on the context. The shift may result from a positionning error like in (c) or can be intentionnal like in (d) where the artist Kruusämagi used the shift to produce white spaces representing snow: white lines are thus *part* of the graphic and are expected on the vectorized version of the artwork.

2.3. Texture and text

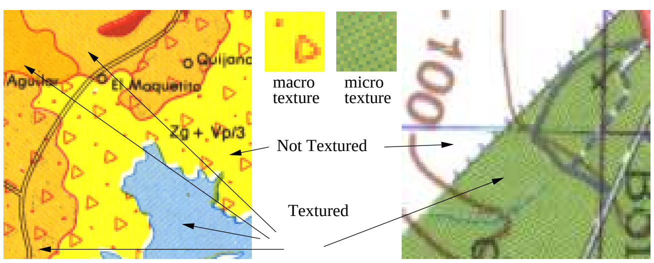

Figure 2. Micro and macro texture

Consider the bitmaps displayed in Fig 2. Some regions present a micro-texture, others a macro-texture (a repetition of triangles), and some present both. A basic R2V tool should recognize regions with a micro-texture as a whole, while the macro-texture and the text should be isolated in corresponding colored layer.

3. R2V processing

The first step of a R2V processing is thus to identify the graphic colors, then label the pixels according to their colored layer, and finally approximate contiguous pixels with lines and regions.

3.1. Preprocessing

Graphics are designed by humans for humans; a transformation to the L*a*b* space will thus guarantee a better color separation. In order to ignore border pixels while keeping lines, we build a mask M by thresholding the norm of the gradient of the luminance and adding ridges [Lacroix and Acheroy, 1998].

3.2. Clustering

Clustering is performed by the "median-shift", so called by analogy with the mean-shift[2] [Comaniciu and Meer, 2002], an iterative procedure that shifts each data point towards the median of data points in its neighborhood . Though the median computation in itself is more expensive than the mean, the convergence of the algorithm is much faster and is less sensitive to outliers.

Let the data be a set of points $\mathbf{x}_i$ embedded in a n-dimensional Euclidean space : $\mathbf{x}_i = (x_{1i}, x_{2i}, ..., x_{ni})$ and E be an n-ellipsoid characterized by its half axis $\mathbf{r} = (r_1, r_2, ...r_n)$ and S_i, the set of data points $\mathbf{x}_j$ embedded in the ellipsoid centered on $\mathbf{x}_i$. Then, $|x_{ki} - x_{kj}| < r_k$ for $k = 1, ..., n$

The algorithm proceeds as follows. At each iteration, all data points $\mathbf{x}_i$ of the full data point set are processed in parallel: the data points $\mathbf{x}_j$ belonging to the ellipsoid S_i centered on $\mathbf{x}_i$ are considered for the median computation, "shifting" $\mathbf{x}_i$ towards the median point $\mathbf{m}_i$ which will be considered for the next iteration, where $\mathbf{m}_i = (m_{1i}, m_{2i}, ..., m_{ni})$ such that m_{ki} is the median of the kth component of all points in S_i. The process intuitively converges, but this remains to be proved. In our experiences, the convergence is much faster than for the mean.

The median-shift algorithm is thus used in the L*, a*, b* parameter space for all pixels belonging to the mask M. In our implementation we used a fixed pixel neighborhood. The image is subdivided in buckets (typically 64×64), so that any pixel of this bucket has the bucket as neighborhood. Only the clusters that contain enough pixels are kept. Then the median-shift algorithm is used again to cluster the results of all buckets. The output of the module is a set of s cluster centers noted $\mathbf{c}_i = (c_{1i}, c_{2i}, ..., c_{ni})$ where $i = 1, ..., s$ such that for any $i \neq j$ there exist at least one k such that $|x_{ki} - x_{kj}| \geq r_k$.

3.3. Labeling

In [Comaniciu and Meer, 2002], a pixel receives the label of the cluster it converges to. Instead, we recommend a labeling process using r_i and the pixel type such as the fuzzy rules having a profile depending on the pixel type as shown in Fig 3.

The process starts with the pixel-line, then their neighbors, then proceeds with all other non-edge pixels, and finishes with edges, using rules that depends on their type. The pixel

2. There exist many variants of the mean-shift, but according to the author's knowledge, none is using the median instead of the mean

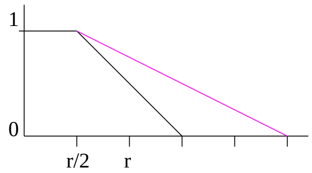

Figure 3. Two fuzzy rules.

is finally assigned to the cluster for which it has the highest membership function. If the membership function to each cluster center is zero, the pixel recieves an undefined label. The rule is more strict for pixel line (dark line in Fig 3 than for pixel edge (pink line in Fig 3), as the latter may be influenced by the neighbouring region.

3.4. Vectorization

The labeled image is separated in each layers and morphology operations are used to fill regions. For lines, the distance to each cluster is computed, and a dark line detection is computed on this distance image [Lacroix and Acheroy, 1998]. Lines containing enough pixels corresponding to the current layer are retained. A regularization step is then necessary to obtain smooth lines and uniform regions.

4. RESULTS

The mean-shift, median shift (with/without mask M) and VectorMagic[3] are compared. The mean and median implementation are the same except for the mean/median computation.

Cluster colors identified on the bitmaps shown in Fig 4 and 7 are displayed in Fig 6 and some labeling details on Test 1 are shown in Fig 5. In Fig 6 note the absence of white in the mean-shift clustering when M is not used. Blue seems corrupted by violet which is not present, and seems to have moved to pink. Neither yellow nor dark brown are present. The version using the mask ("no-edge") is slightly better. The median-shift clusters with M is about the same as without, but only one violet cluster is found, which is better. All important colors of the image are represented except that there is no yellow green (close to yellow). Vector-Magic proposes several palettes; we have chosen the one with 12 colors. Note that one gray is corrupted by blue, while no blue cluster is provided. There is no yellow, but a yellow-green.

Details of the labeling are shown in Fig 5. Dark blue indicate unlabeled pixels. Regularization of Vector-Magic (VM) is superior, but many colors are better assigned with the proposed labelling based on the median-shift clustering

3. "VectorMagic" (http://vectormagic.com/) is a free web service (only labeled rasters are available for free) considered as one of the best vector-to-raster software

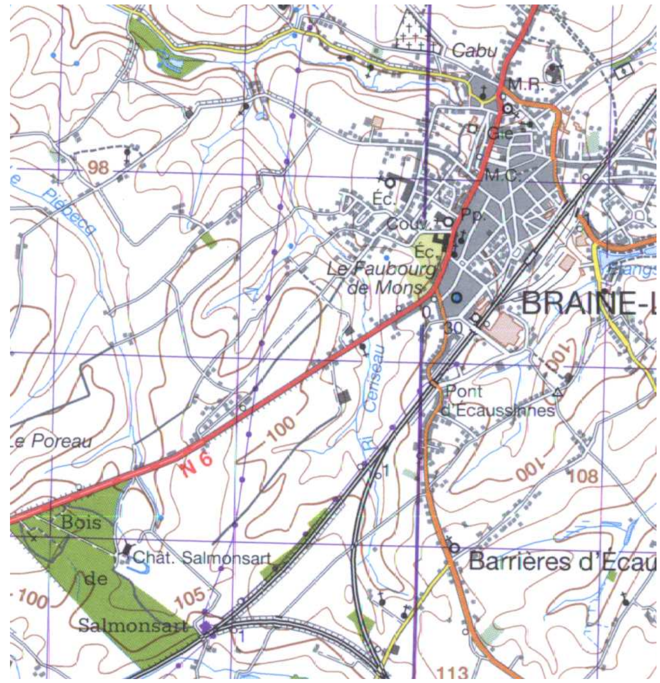

Figure 4. Test 1.

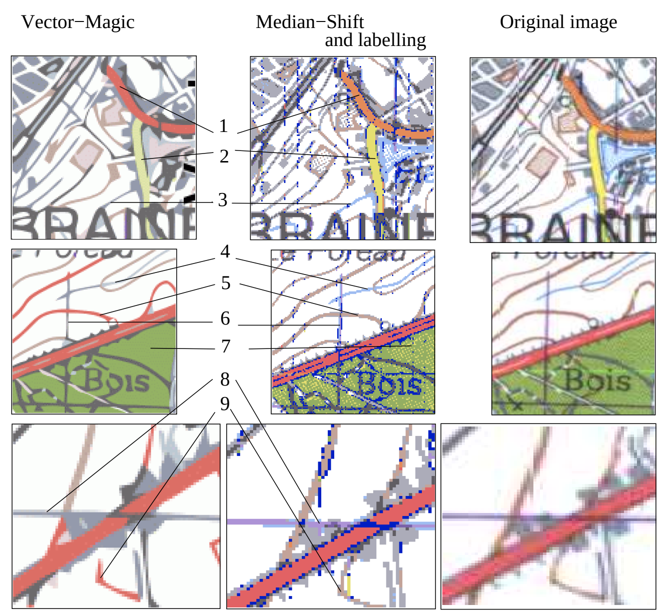

Figure 5. Parts of Labeled Test 1.

(MSC). For example in (1) MSC correctly found the orange of the road while VM found red; in (2) it correctly found yellow while VM found green; in (3) MSC mostly assigns brown to contour lines while VM often assigns them to grey or to red like in (5) and (9); in (3) and (4) the river is correctly assigned to blue by MSC while it is set to grey in VM; finally violet lines identified as such by MSC and not by VM as shown in (6) and (8). On the other hand VM deals better with texture as shown in (7).

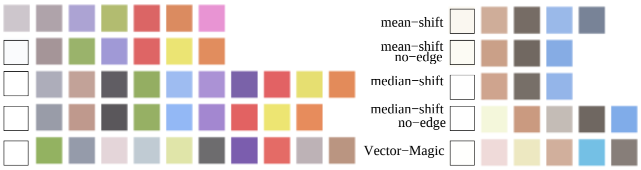

Figure 6. Cluster colors provided by different algorithms on Test 1 (left) and Test 2 (right).

For the second test, where the parameters where kept the same, the mean-shift and the version of the median-shift without pre-processing are missing the green cluster. The blue cluster seems better identified in the median-shift. No clustering method was able to isolate the shading tone.

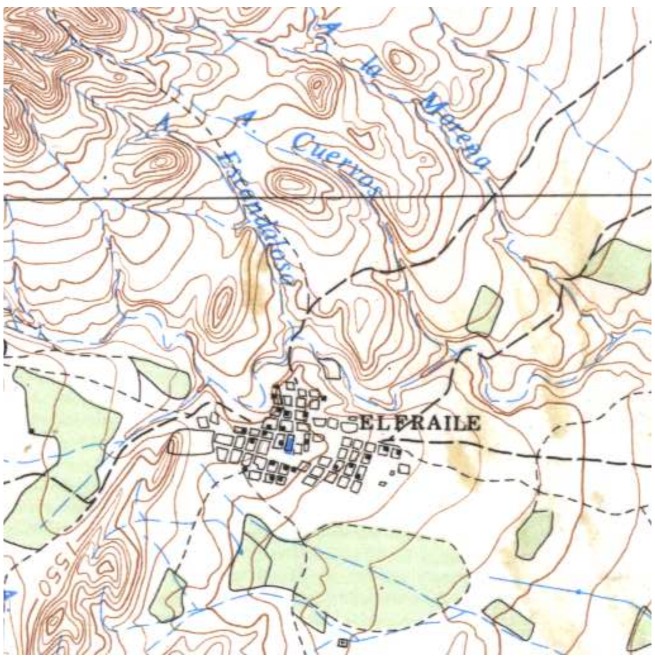

Figure 7. Test 2.

5. Conclusions

Some problems of R2V conversion have been analyzed, and a strategy has been proposed. The latter involves a pre-processing generating a mask from which edges are removed and lines kept. A clustering is then performed while considering only the pixels of the mask. A new algorithm, the median-shift, has been proposed in this context. Then comes the labeling process which should also take the pixel type into account. The vectorization is then performed on each layer, but, instead of considering a binary image, the distance to the cluster is used, enabling the extraction of dark lines for a better line vectorization. The last step involves a regularization procedure.

The importance of the pre-processing ignoring edge pixels while keeping lines has been shown on some examples. Tests also showed the superiority of the median-shift over the mean-shift, and over the the clustering method used by Vector-Magic.

Efforts are still necessary for the labelling regularization and for handeling textured areas.

References

[Chen et al., 2006] Chen, Y., Wang, R., and Qian, J. (2006). Extracting Contour Lines From Common-Conditioned Topograpic Maps. *IEEE Transactions on Geoscience and Remote Sensing*, 44(4):1048–1057.

[Chhabra and Philips, 2000] Chhabra, A. and Philips, I. (2000). Performance evaluation of line drawing recognition systems. In *Proc. 15th Int'l Conf. Pattern Recognition*, volume 4, pages 4864–180)4869.

[Comaniciu and Meer, 2002] Comaniciu, D. and Meer, P. (2002). Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24(5):603–619.

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

[Hilaire, 2006] Hilaire, X. (2006). RANVEC and the Arc Segmentation Contest: Second Evaluation. In Liu, W. and Lladós, J., editors, *Graphics Recognition—Ten Years Review and Future Pespectives*, volume 3926 of *Lecture Notes in Computer Science*, pages 362–368. Springer Verlag.

[Hilaire and Tombre, 2006] Hilaire, X. and Tombre, K. (2006). Robust and Accurate Vectorization of Line Drawings. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(6):890–904.

[Lacroix and Acheroy, 1998] Lacroix, V. and Acheroy, M. (1998). Feature-extraction using the constrained gradient. *ISPRS Journal of Photogrammetry and Remote Sensing*, 53(2):85–94.

[Pouderoux, 2007] Pouderoux, J. (2007). *Création semi-automatique de modèles numériques de terrains - Visualisation et interaction sur terminaux mobiles communicants*. PhD thesis, Université de Bordeaux.

[Robert, 1997] Robert, R. (1997). *Contribution la lecture automatique de cartes*. PhD thesis, Université de Rouen, Rouen, France.

[Tombre, 2006] Tombre, K. (2006). Graphics Recognition: The Last Ten Years and the Next Ten Years. In Liu, W. and Lladós, J., editors, *Graphics Recognition—Ten Years Review and Future Pespectives*, volume 3926 of *Lecture Notes in Computer Science*, pages 422–426. Springer Verlag. Report from final panel discussion at GREC'05.