Optimal palette extraction as part of scanned graphics vectorization

Vinciane Lacroix and Mahamadou Idrissa
Royal Military Academy, Brussels, Belgium

Final Objective: Automatic raster to vector transformation of scanned graphics; application to scanned maps
Partial Objective: Automatic raster to vector transformation of scanned graphics; application to scanned maps

Final Objective:
Automatic raster to vector transformation of scanned graphics; application to scanned maps

Partial Objective:
The ideal still basics tool should extracts points, lines and regions together with:
- A list of points p location
- A width
- A color of border
- A color of region
- A filled? style
- A region: uniform, cf line
- Specific shapes (circle, piecewise cubic, etc.)

Table 1 Some basic yet ideal vectorial representation

<table>
<thead>
<tr>
<th>Features</th>
<th>attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>location (x, y)</td>
<td>x: pt of points; y: border/closed line</td>
</tr>
<tr>
<td>color</td>
<td>Color: R, G, B (x) / R or G or B (y)</td>
</tr>
<tr>
<td>style</td>
<td>continuous, dashed, dotted</td>
</tr>
<tr>
<td>region</td>
<td>style of border: region uniform, textured, gradient</td>
</tr>
<tr>
<td>description</td>
<td>c.n. circular, polygonal, ellipse, circle/elliptic arc, specific shape (circle, square, rectangle, etc.)</td>
</tr>
</tbody>
</table>

Overall proposed strategy

- Optimal palette extraction: \( \mathbb{P} = \{p_1, \ldots, p_n\} \) the “representative” colors of all image features.
- Expected problems: colors missed (too close colors), colors added (color at edges, color superposition, mixed pixels, noise), wrong colors (thin features).
- Image Labelling: gives \( \mathbb{L} = \{l_1, \ldots, l_n\} \) associated to the colors of \( \mathbb{P} \), and \( f_l \), the best labeled image that gave rise to the original bitmap \( f \).
- Expected problems: missing feature labels, wrong labels at features superposition and at edges.
- Connected components extraction: recovering points, lines and regions together with their attributes.

This poster concentrates on the first part of the process and gives some hints on the second part.

Some difficulties of palette extraction are shown in these examples. Image A and B have the same pixels but should have different palettes. Image C (a photograph of a Macbeth color chart [2]) should have a palette of about 25 colors, but its extraction seems easy when seen in true-colors but less easy when seen in pseudo-colors.

Color quantization:
- aims at coding the image with \( n \) colors
- Note: pixels at edges requires intermediate colors

Clustering:
- identifies in the parameter space (RGB, L*a*b*, etc.) \( n \) “representing” colors associated to each cluster of data
- Note: constraints on clusters such as their number, shape (hyper-sphere + radius), separating planes, etc.
- Both ignore spatial information except mean-shift (color + space)

Top: Optimum palette extraction. On both images, zooms of the proposed paletteization, of original and of GIMP palettization are shown on the left, middle and right respectively. Middle: palettes extracted by all methods. Bottom: Regularization of the labeled images

Proposed method for palette extraction

A map \( 64 \times 64 \) cm by 40 cm \( \rightarrow \) 10078 pb by 6299 pb (600 dpi), \( \rightarrow \) 4 random 512×512 zones
- extract 512×512 zones, for each zone:
  - extract colors of “uniform” regions with median-shift.
  - extract non-yet discovered colors of features (line + points)

Median-shift [1]

Iterative: shifts a point towards the “median” of its neighborhood, median: has as components the median of each component; neighborhood: all colors less distant than \( \delta \), distance \( \delta \) in a range of [16-19] (minimum error).

Evaluation of palette extraction

- qualitative:
  - visually judge the palette extracted from image I or
  - visually compare the image I and \( I' \) if \( I' \) is such that the color of each pixel of \( I' \) is replaced by the nearest color of the palette, or by the color representing the cluster it belongs to.
- quantitative:
  - compute a distance \( D \) between I and \( I' \)
  - \( D(I',I) = \sum E(l_i) / N \)

where \( E(l_i) = \max \) Euclidean distance in Lab and CMC color difference.

Table 2 Mean error for several images and algorithms

Conclusions

For optimum palette extraction, we recommend a two-phases strategy. During the first phase uniform colors are extracted using a clustering algorithm, then the non yet discovered colors of features are sought. The median-shift in Lab color space with the maximum of the Euclidean and the CMC distance gives excellent qualitative and quantitative results on all images except one image. The quantitative tests however do not always reflect the perception, so that another distance than the sum of the distance on all pixels should be proposed.

References


Contact information:
Vinciane Lacroix, Department of C4I, Royal Military Academy, Av de la Renaissance 90, 1090 Brussels, Belgium.
Email: vinciane.lacroix@defr.bne.be
This work has been funded by the Belgian Ministry of Defense as part of the C4’12 study.