

Using Color Information as a Whole in Image Segmentation

Rodolfo Alvarado Cervantes and Serguei Levachkine

Geoprocessing Laboratory
Centre for Computing Research (CIC) - National Polytechnic Institute (IPN)
Mexico City, Mexico
rodolfox@mexico.com sergei@cic.ipn.mx

Abstract. In this work a semiautomatic image segmentation method is presented. It uses the color information for each pixel as a whole avoiding information scattering. At present hundreds of segmentation techniques are available for color images, but most of them are monochromatic methods applied on the color components of different color spaces and then combined the results in different ways. The problem is, when the color components of a pixel are processed separately, the color information is so scattered in its components that the results are often useless for the subsequent (higher level) image processing. Our method aims at solving this problem. The improvement in quality of our segmentation technique is notorious.

1 Introduction

Image segmentation consists of partition or separation of the image in different regions, which are homogenous in some characteristic. It is an important feature of human visual perception, which appears spontaneously and naturally. It is also the first and most important task in image analysis and processing [1][3][4][8][9]; all subsequent steps like features extraction and recognition greatly depend on segmentation quality. Without a good algorithm, objects of interest cannot be recognized in slightly complex images and the whole image processing system will fail [1][2][5][9][10]. Therefore considerable effort is taken in the design of algorithms that increment the probability of a successful segmentation. At present hundreds of segmentation techniques are available for color images, but most of them are just monochromatic methods applied on the components of different color spaces and then combining the results in different ways. The problem is, when the color components of a pixel are processed separately, the color information is so scattered in its components that the results are often useless for the subsequent (higher level) image processing [4][5][6][7][9][10].

In this work a semiautomatic image segmentation method is presented, which as opposed to the previous work, uses the color information for each pixel as a whole avoiding the mentioned information scattering. In this method the three color components are combined in two parts: in the definitions of colors distances $[\Delta_r, \Delta_g, \Delta_b]$ and in the definition of the “Color Probability Images”.

The method is primarily based on the concept of the “Color Probability”, i.e. the probability that a pixel has the same color, which a user has previously selected. This color probability is calculated for every pixel with a formula developed for this

purpose to build Color Probability Images (CPI's), which can, if needed or desired, be processed with any tool of mathematical morphology for gray images combining color and geometrical information in a simple way.

The improvement of the segmentation quality is quite notorious and can be appreciated on results shown in section 3 where this method is compared with one, state-of-the art commercial software, using scattering type segmentation algorithms (the only available today in literature).

2 Description of the method

Our technique can be conceptually divided in two steps:

1. The generation of color probability images (CPI's) by solving equation (1) for each pixel in the input color image. Only color information is used in this step resulting in a pixel-oriented technique.
2. The application (if necessary) of any desired morphology technique for gray images on the previously generated CPI's. In this way geometric discriminant characteristics are introduced in the segmentation process.

By the use of the concept of color probability, the color information of every pixel is processed as a whole without separating the color components. To generate a CPI is needed:

1. A color image in RGB format.
2. A set of pixels forming a sample of the color desired

From the sample we calculate statistical values according to a HSI modified model (section 2.6). To obtain a CPI we calculate for every pixel in the color image the color probability $CP(i,j)$:

$$CP(i,j) = e^{\left(\frac{-a_1\Delta_h^2}{2\sigma_h^2}\right)} * e^{\left(\frac{-a_2\Delta_s^2}{2\sigma_s^2}\right)} * e^{\left(\frac{-a_3\Delta_i^2}{2\sigma_i^2}\right)} \quad (1)$$

$e = 2.7182...$

$\Delta_h = \text{Hue_distance_}\Delta_h(\text{hue}(i,j), \text{average_hue})$

$\Delta_s = \text{saturation_distance_}\Delta_s(\text{saturation}(i,j), \text{average_saturation})$

$\Delta_i = \text{intensity_distance_}\Delta_i(\text{intensity}(i,j), \text{average_intensity})$

$\sigma_h = \text{Hue standard deviation}$

$\sigma_s = \text{Saturation standard deviation}$

$\sigma_i = \text{Intensity standard deviation}$

$[a_1 \ a_2 \ a_3] = \text{Mask of refinement, } a_i \text{ can be 0 or 1}$

Some modifications on standard HSI color space were necessary in order to create a consistent model to represent color and color centroids:

1. *Representation of hue.* Instead of standard representation of hue as an angle between $[0 \ 360]$, hue is represented as a normalized vector in R^2 (with magnitude 1 or 0). This representation has at least 3 advantages compared to an angle $[0 \ 360]$: a) The existing discontinuity in 360 and 0 degrees is eliminated. b) Hue average of a group of pixels can be understood as the resulting angle of the vector addition of the pixels in the chromatic region of the sample, giving a simple manner to calculate the average. c) Setting magnitude to 0 or 1 works as a flag intended for distinction between chromatic or achromatic regions.
2. *Separation of chromatic and achromatic regions of the HSI space.* We use a separation of the region (according to [1]) with possibility of refining with the $[a_1 \ a_2 \ a_3]$ mask (in equation (1)) in order to calculate the hue average and Δ_h . Once calculated Δ_h , Δ_s and Δ_i this distinction is not longer necessary because in the formulation of CP (equation (1)) all the cases of color comparison between zones are accounted and maintain consistency.

2.1 Hue average calculation

In order to obtain a value of hue, which represents the average of several pixels of a sample, we take advantage of the vector representation in R^2 ; the vectors that represent the hue values of individual pixels are combined using vector addition. From the resulting vector we obtain two significant values used in the algorithm: average hue and relative saturation. Thus H_m is calculated in this manner:

1. For every pixel in the sample the following R^3 to R^2 transformation is applied:

$$VR(P) = \begin{cases} \begin{bmatrix} 1 & -\cos(\pi/3) & -\cos(\pi/3) \\ 0 & \sin(\pi/3) & -\sin(\pi/3) \end{bmatrix} * \begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} & \text{If } P \notin G \\ \begin{bmatrix} 0 \\ 0 \end{bmatrix} & \text{If } P \in G \end{cases}$$

G represents the achromatic zone in the HSI space. **[RGB]** are the color components of the pixel in the RGB color space

2. Equivalent to the following pseudocode is executed

```

Vector.x = 0;
Vector.y = 0;           // initialize vector
For (i = 1; i <= n; i++) // for every pixel in
the sample do
{
    Vector.x = Vector.x + VR(i).x;

```

```
Vector.y = Vector.y + VR(i).y;}
```

In this code we have a vector, which accumulates the vector additions as index i increments. Each of the vectors being added corresponds to the previous R^3 to R^2 transformation for every pixel in the sample made in step 1.

3. The angle respect to the X-axis that is obtained corresponds to average: $H_m = \text{angle}(\text{Vector}, 0)$.
4. Relative saturation S_r is calculated: $S_r = \text{Magnitude}(V_s) / \text{Magnitude}(V_1) + \text{Magnitude}(V_2)$

2.2 Achromatic region G

The achromatic zone **G** is the region in the HSI color space where no hue is perceived. That means color is perceived as gray levels because color saturation is very low or intensity is either too low or too high.

Given the three-dimensional HSI color space, we define the achromatic region **G** as the union of the points inside the cylinder defined by $\text{Saturation} < \text{threshold}_1$ and the two cones $\text{Intensity} < \text{threshold}_2$ and $\text{Intensity} > \text{threshold}_3$. Pixels inside this region are perceived as gray levels (Fig. 1).

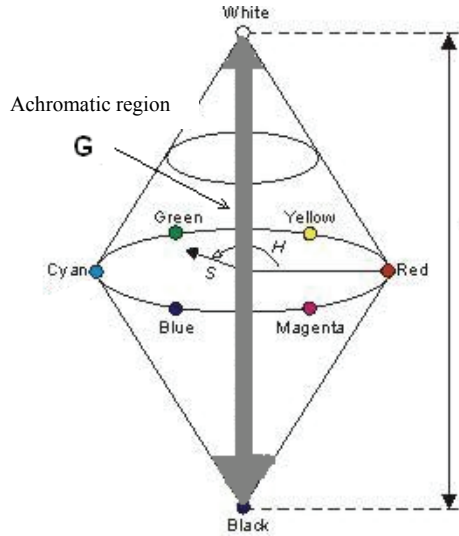


Fig. 1. Achromatic region **G**

2.3 Hue distance Δ_h

Using the vector representation of Hue, we define hue distance Δ_h for two pixels P_1 and P_2 as follows:

$$\Delta_h(P_1, P_2) = \begin{cases} K * \text{abs}(\arccos[\text{dot_product}(V_1, V_2) / \text{mag}(V_1) * \text{mag}(V_2)]) & \text{If } P_1 \text{ and } P_2 \notin \mathbf{G} \\ 0 & \text{If } P_1 \text{ or } P_2 \in \mathbf{G}. \end{cases}$$

$\text{abs}()$ = absolute value

$\arccos()$ = arc-cosine

$\text{mag}()$ = magnitude of vectors

$\text{dot_product}()$ = vector dot product

K is a normalizing real factor to force $\Delta_h \in [0, 1]$.

$\mathbf{G}$ is the achromatic region

V_1 y V_2 are the vectors in R^2 calculated with the transformation on P_1 and P_2 :

$V_m(P) =$

$$\begin{bmatrix} 1 & -\cos(\Pi/3) & -\cos(\Pi/3) \\ 0 & \sin(\Pi/3) & -\sin(\Pi/3) \end{bmatrix} * \begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} \quad \text{If } P \notin \mathbf{G}$$

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \text{If } P \in \mathbf{G}$$

2.4 Saturation distance Δ_s and intensity distance Δ_i

Saturation distance and intensity distance between two pixels is defined as:

$$\Delta_s = \text{abs}[\text{saturation}(P_1) - \text{saturation}(P_2)] \text{ and } \Delta_i = \text{abs}[\text{intensity}(P_1) - \text{intensity}(P_2)]$$

2.6 Statistical values for groups of pixels

The statistical values needed in equation (1) are calculated as follows:

$$\text{Saturation average} = \frac{\sum_{i=0}^n \text{saturation}(i)}{n}; \quad (1)$$

$$\text{Intensity average} = \frac{\sum_{i=0}^n \text{intensity}(i)}{n} \quad (2)$$

$$\text{Hue standard deviation } \sigma_h = \sqrt{\frac{\sum_{i=1}^n \Delta_h^2(i)}{n-1}}; \quad (3)$$

$$\text{Saturation standard deviation } \sigma_s = \sqrt{\frac{\sum_{i=1}^n \Delta_s^2(i)}{n-1}} \quad (4)$$

$$\text{Intensity standard deviation } \sigma_i = \sqrt{\frac{\sum_{i=1}^n \Delta_i^2(i)}{n-1}} \quad (5)$$

n is the number of pixels of the sample.

Δ_h = hue_distance_ Δ_h (hue(i,j), hue_average)

Δ_s = saturation_distance_ Δ_s (saturation(i,j), saturation_average)

Δ_i = intensity_distance_ Δ_i (intensity(i,j), intensity_average)

2.7 Use of mathematical morphology on CPI images

The CPI image represents the image pixels that has the maximal probability of belong to the previously selected color. As this image is a gray image, can be treated with any tool of mathematic morphology for gray images. Filters, operators, thresholds, etc can be applied directly to the CPI strengthening the segmentation as geometrical characteristics are introduced. The common intensity image can be processed too as a complementary information source.

3 Experimental results

Our segmentation method was applied to a section of a topographic map (See Fig. 2)

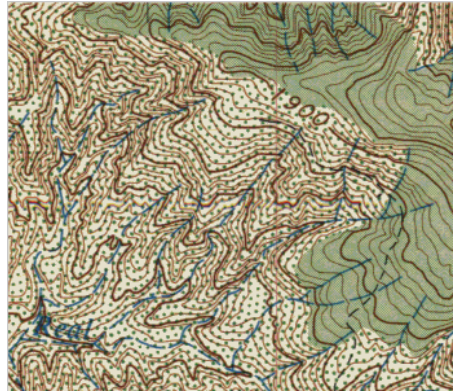


Fig. 2. Section of a topographic map

The thematic layers contained in the image are:

1. River layer (blue lines)
2. Gross brown isolines
3. Thin light brown isolines
4. Green area

1. River layer (blue lines)
2. Gross brown isolines
3. Thin light brown isolines
4. Green area
5. Green spots

To segment the river layer we took a pixel sample, which looks representative at a simple view (See Fig. 3)

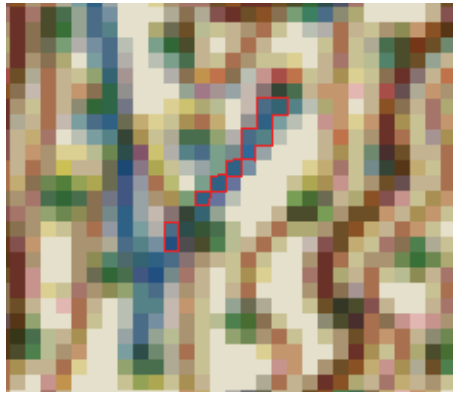


Fig. 3. Sample of blue pixels of the river layer

2. Using this sample we calculated the color centroid and deviation:

Color centroid $\{H_c \in [0, 2\pi], S_c \in [0, 1], I_c \in [0, 255]\} = (3.56, 0.4126, 102)$

$\sigma_{ti} = 0.0392$; $\sigma_{si} = 0.0884$; $\sigma_{ii} = 0.0382$

3. We generated the corresponding CPI (Fig. 4)



Fig. 4. CPI of blue pixels

4. We applied a simple threshold to this image (Fig 5) to segment the layer. The range for values of thresholds is very wide (similar results were obtained for range [10 150]) we choose value equal to 30:

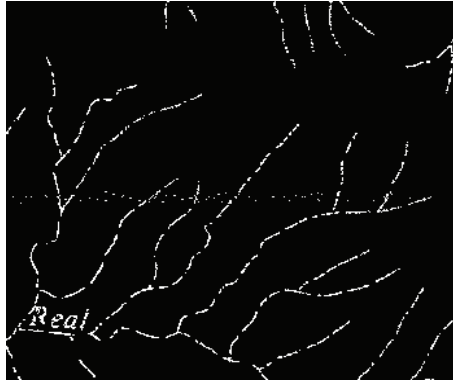


Fig. 5. Result for threshold of 30 on CPI

We observed that even with a coarse selection of the pixel sample, we obtain a good separation of this layer. Several different pixel samples were taken varying the number of pixels from 3 to 10; all of them gave very good results.

Looking at Fig. 2 we can observe that color is a discriminant characteristic of the river layer. That means no other layer has a color, which a human can confuse with this river layer. That is why we can segment it so easily only with the threshold (with a wide range) of corresponding CPI. This doesn't happen with the other layers; so additional morphological processing is needed. In some cases already extracted layers are necessary to extract some others (case of Fig. 12).

Fig 6 can be considered very acceptable as input for recognition, but we can refine it in this way: Apply dilatation followed by elimination of small disconnected areas. The result is shown in Fig 7.

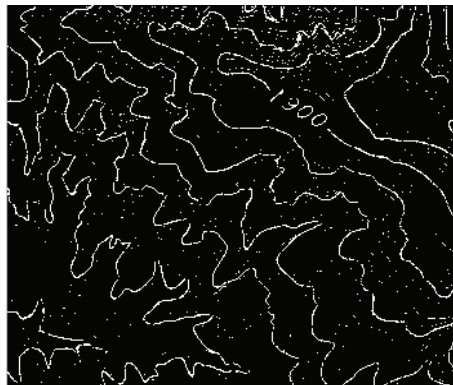


Fig. 6. Result for threshold of 80 on CPI

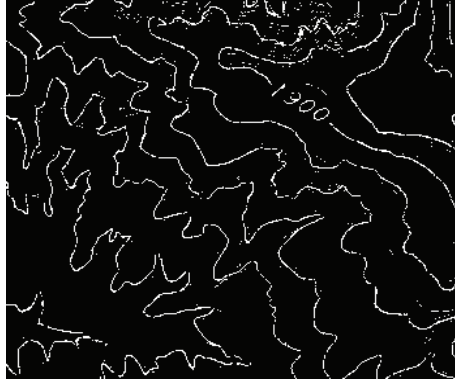


Fig. 7. Segmentation of gross brown isolines

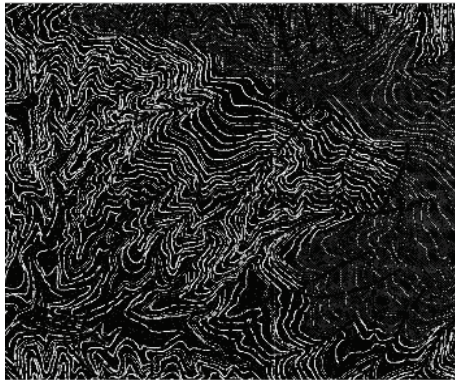


Fig. 8. CPI image of thin light brown isolines

To improve the quality of Fig. 9 we subtract the dilatation of Fig 7 to eliminate any confusion with that layer. The result is shown in Fig 10. For the green area and the green spots we present the final results (Fig. 11 and Fig.12).

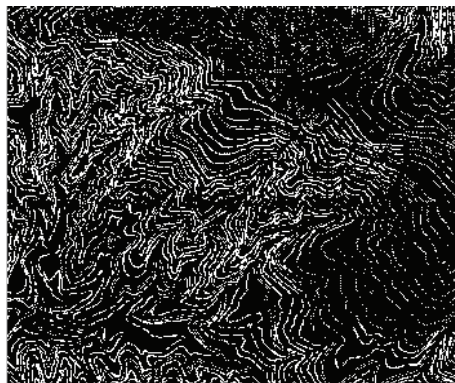


Fig. 9. Result for threshold of 150

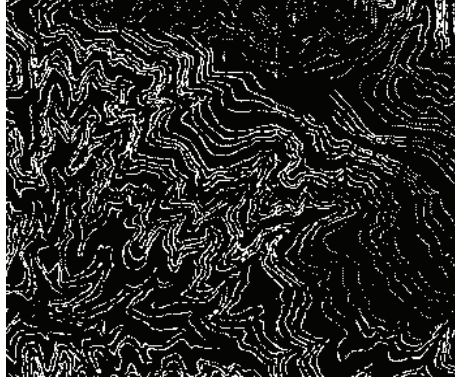


Fig. 10. Segmentation of thin light brown isolines

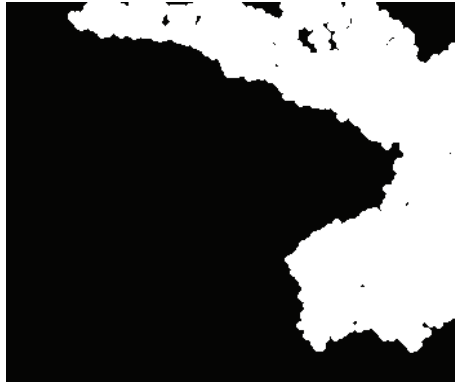


Fig. 11. Segmentation of green area

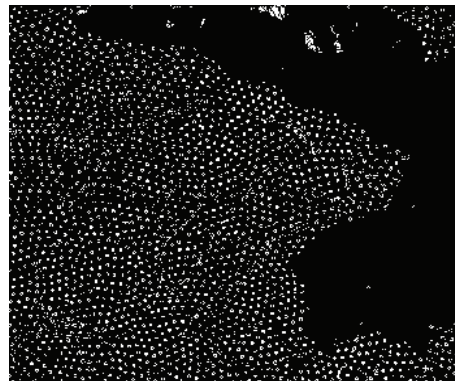


Fig. 12. Segmentation of green spots

Now we process the same map using the software R2V from Able Corp. This software additionally to segmentation performs line recognition, so not all the points shown are obtained from the segmentation step (See Fig 13, 14, 15, 16, 17).



Fig. 13. Segmentation of river layer

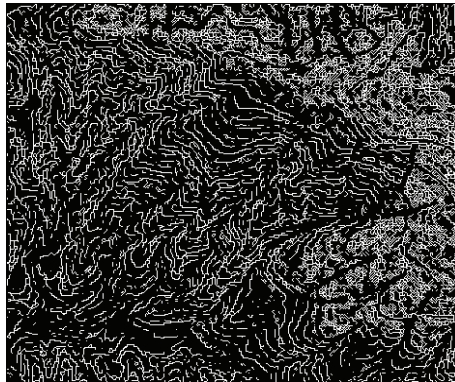


Fig. 14. Segmentation of gross brown isolines

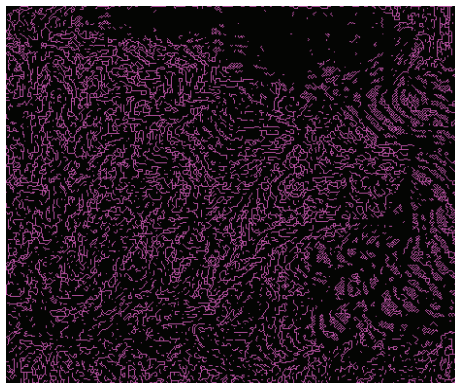


Fig. 15. Segmentation of thin light brown isolines

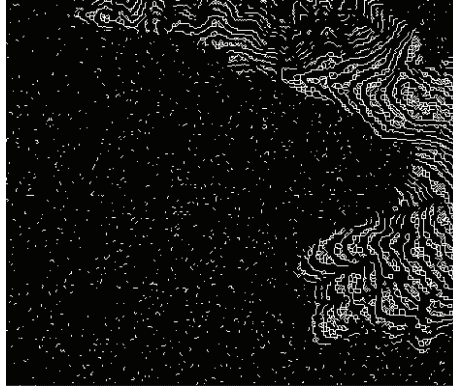


Fig. 16. Segmentation of green spots

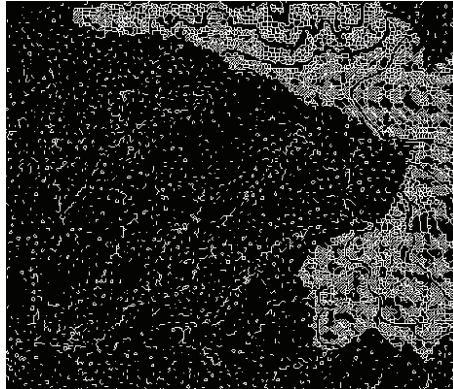


Fig. 17. Segmentation of green area

After comparing the results of our system with those from R2V, we can observe that in R2V there are a large number of pixels that should appear in the segmentation of the layer but they do not. The worst case is in Fig. 13 that is practically empty. We can also observe in R2V that there is a big amount of confusion of pixels from other layers for example in Fig. 14 segmentation of gross brown isolines, there appears many pixels from the green area that should not, and in Fig. 15 appears pixels from all layers.

4 Conclusions

From an analysis of the results we can state that the method exposed in this paper offers a useful and efficient alternative for the segmentation of color images. We cannot say the same about R2V because none of the results is even close in quality to the thresholded CPI's. This gives us a clue that in R2V the color information is not

well used as in a CPI. From experiments in many different maps we can mention some advantages of our technique:

1. *Simplicity*. Required steps to obtain a good segmentation of the layer of interest are usually simple and repetitive. If color is a discriminative characteristic of the layer of interest, only threshold of the CPI is needed to obtain a good segmentation.
2. *Very low computational complexity*. The generation of a CPI only requires solving equation (1) for every pixel. Thus the complexity is linear with respect to the number of pixels of the source image.
3. *Layer separation by color*. Generally other methods extract punctual and linear objects together, making necessary much additional work to separate those layers
4. *Good segmentation of layers using color only*. From many experiments we observed that a good percentage of layers were obtained from threshold of CPI only.

References

1. Plataniotis, K.N., A.N. Venetsanopoulos: Color Image Processing and Applications, 1st Edition Germany, Springer, 354p
2. Alvarado Cervantes, Rodolfo, Master Thesis: Segmentación de patrones lineales, topológicamente diferentes, mediante agrupamientos en el espacio de color HSI, Centre for Computing Research (CIC) - National Polytechnic Institute (IPN), Mexico 2006
3. Serguei Levachkine: Raster to Vector Conversion of Color Cartographic Maps. GREC 2003: 50-62
4. Serguei Levachkine, Efrén González-Gómez, Miguel Torres, Marco Moreno, Rolando Quintero: Knowledge-Based System for Color Maps Recognition. KES (1) 2005: 297-303
5. Angulo, Jesús, Serra Jean : Mathematical morphology in color spaces applied to the analysis of cartographic images, GEOPRO 2003: 59-66
6. Efrén González-Gómez, Serguei Levachkine: Color Cartographic Pattern Recognition Using the Coarse to Fine Scale Method. CIARP 2004: 533-540
7. Serguei Levachkine, Miguel Torres, Marco Moreno, Rolando Quintero: Knowledge-Based Method to Recognize Objects in Geo-Images. KES 2004: 718-725
8. H.D. Cheng, H.D. Jiang, Y. Sun & Jingli Wang: Color image segmentation: advances and prospects, Pattern Recognition 2001, 34(12), 2259-2281
9. Hanbury, Allan; Serra, Jean: A 3D-polar Coordinate Colour Representation Suitable for Image Analysis, Technical Report PRIP-TR-77 Austria 2002
10. Serguei Levachkine, Miguel Torres, Marco Moreno, Rolando Quintero: Simultaneous Segmentation-Recognition-Vectorization of Meaningful Geographical Objects in Geo-Images. CIARP 2003: 635-642