# Segmentation of Climatological Graphic Records

Matheus Sant Ana Lima[1]

UFSCar, Federal University of São Carlos, São Carlos, Brazil,
Computer Departament
Center of Exact Sciences and Technology
matheus_santana@comp.ufscar.br

**Abstract.** This paper is focused in the segmentation problem of the continuous curve traces records in the anemograms made by the detection of instantaneous values of wind speed(*gusts*) on continuous chart. These records are made by several many years using paper records, consequently the analysis of these data's need considerable human effort. The available literature of segmentation in climatological chart's are very scarce, thus, this paper propose a model to help this problem. The method present is based in the *clusters* analysis of the input image, that is used for classify the object's in distinct groups: curve traces, scale-grid and background; followed by noise filtering and encoding of the geometrical configuration of the curve traces. Since it's necessary just maximum and minimum points for represent the respective *f(x)* function of the curve trace, is proposed an algorithm that mark the correspondent pixel's for such values. It's also discuss the experimental results from the implementation of the proposed method.

**Keywords:** cluster segmentation and grouping, anemogram, shape representation and analysis, document analysis, wind speed chart

## 1 Introduction

The register of climatological data, in general cases, is a very complicated process for many reasons, like human effort, large ammount of paper documents produced daily, considerable time dispended of manual data analysis, storage and organization. The segmentation and extraction of relevant information with digitalized papers registers can improve significantly the use of the information contained inside it. The climatological register considered in this paper covers anemographs specialy the values of wind speed, were provided by the Climatological Station basead in Federal University of São Carlos, Brazil, operated by Nacional Institute of Meteorology(INMET). Allaby [8] descrive the basic structure of a anemometer:

> "*The rotating cups anemometer is the device that is most widely used. It consist of three or four hemispherical or conical cups that are mounted on arms separeted by 120°or 90°, depending on the number of cups, and attached to a vertical axis. The wind exerts more pressure on the concave*

*surfaces than on the convex surfaces. This causes the cups to turn about the axis, and the rotational speed of the axis is converted into the wind speed and displayed on a panel attached below the cups or remotely, by a needle on a dial."*

These research can be applied to support project planning on engineering and architecture, calculation of metallic structures and thermal comfort in buildings, management and control of risk areas by governments, flight planning at airports and many others situations.

In fig.1 is a reprodution of an parcial chart of wind speed from an anemogram used in this experiment. It's possible to see the curve trace often seen hard to

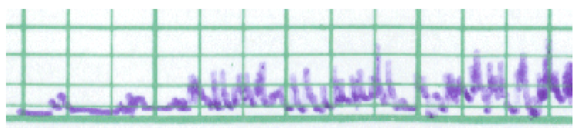

**Fig. 1.** Parcial chart of wind speed extracted from the anemogram record of 02/03/2006

analyze by visual inspection, moreover, at places where the tracing of the curve intersects the grid, the pixel's colors, in RGB standart [2], are very similiar, characterizing the multi threshold behavior [2]. We consider the presence of three clusters of pixels in this kind of image: curve trace, scale-grid and background. The pixels grouped inside each cluster has a very closed similiraty and this allows us to segment the image in the three diferent parts. The cluster analysis based on K-Means algorithm, presented a high quality segmentation, but the scale-grid and background images are unusefull data that are removed by the analysis of each pixel in the image and his 8-neighbors. As result, it's obtained a binary image. The presence of noise after the segmentation was decreased by the average filtering [2,4]. The geometrical configuration of the binary image can be encode using the Freeman method [1], which consists in translating the object inside the image into a 8-diretions sequence, reducing the need of the raster image into a simple sequence of characters. To remove redundant character code, the author utilized graph theory, to find the equivalent sequence without redundante data. We can find the presence of noise in those sequence of characters, also known as Chain Code, caused by previous methods. This noise was reduced by the smooth filtering based on the average of the Chain Code's values sequence.

The corresponding *f(x)* function of the curve traced in the chart can be represented by the maximum and minimum values of the upper and lower profiles, which were extracted from the segmented and encoded object. The main contribuition of the paper is the algorithm to find and mark the important places of the segmented object through concavity analysis. We can get it by studing the values of the lines bounding. Other approaches, such as GILDA [5] use a previous knowledge about the shape and style of the draw lines.

The paper is organized as follows. Section 2 introduces our segmentation framework, describing the methods and tecniques that was used in the experiments. Section 3 demonstrates the experiments results in speed wind graphics contained in a anemogram chart image and Section 4 concludes and suggests future works.

## 2 Segmentation Framework

The techniques used are described next.

### 2.1 Cluster analysis

Each class or clusters of objects in image receive a aproximated value of color in RGB standart [2], based on the euclidian distance, used for the analysis of the similarity between the pixels [10,9].

This methods allows us to classify each pixel in one of the K classes [9]. The choise of the most appropriated initial values for these classes is an unsolved problem in lecture. We use two approach and both have advantages and desadvantages. In the random choise, different executions of a same implementation in a same input image can generate different segmentations and, when the initial values of each class dont represent apropriated the color of the respective object, the quality is compromised. This happens because the colors of the classes converges into innapropriered values. The advantage of this method is the minimum need for human intervention. Fig.2(Left) and fig.2(Right) reproduce different executions using random choise for the initial values of classes from fig.1. In Table 1, C1 and C2 show these values; C3 show the manual choise adopted in fig.1.

| Class(C1) | B | G | R | Class(C2) | B | G | R | Class(C3) | B | G | R |
|---|---|---|---|---|---|---|---|---|---|---|---|
| v1 | 9 | 179 | 171 | v1 | 150 | 164 | 131 | v1 | 167 | 201 | 128 |
| v2 | 67 | 116 | 7.53 | v2 | 63 | 140 | 90 | v2 | 255 | 255 | 255 |
| v3 | 240 | 240 | 240 | v3 | 240 | 240 | 240 | v3 | 212 | 127 | 156 |

**Table 1.** Initial manual and random RGB values.

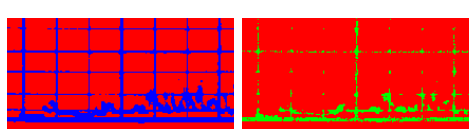

**Fig. 2.** (Left) Segmentation of parcial chart in fig.1 with random initial values C2. (Right) Segmentation of parcial chart in fig.1 with random initial values C1.

The manual choise for the initial values produces better results than randomic methods, but it has as disadvantage the need of human intervention, consequently reducing the efficiency of the system. The K-Means algorithm [9] for clusters analysis used by the author is follow:

**Image segmentation by K-Means algorithm.**

- **Step 1:** Insert the image and boot: the number of $k$ groups (*clusters*), the initial values of $k$ classes and set the stop criterion;
- **Step 2:** Compute the similarity between the pixels of each class;
- **Step 3:** Assign each pixel to the group of most similar class;
- **Step 4:** Calculate the average of each group and assign it to each class according to the groups;
- **Step 5:** Check the stopping criterion, if not satisfied, return to **step 2**.

### 2.2 Removal of neighborhood

The K-means algorithm produces a segmentation appropriated for most of the pixels in the image, it classifies each pixel in its respective class. However some classes represent unwanted information, which can be removed by the the analysis of each pixel and the values of his neighboring pixels. If the majority is classified as background, using window 6, the pixel in question is upgraded to background; if its classified as an object, the pixel is updated as the object. In the case of equality, it is checked using window 8, and if this also fails, the pixel is updated as a background. Fig.3 illustrates the possible neighbors to a pixel $p$. The neighboring pixels marked in red indicate region of object and pixels marked in green as the background region.

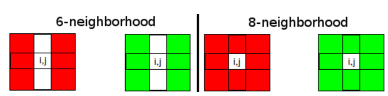


**Fig. 3.** Window of 6-neighborhood and 8-neighborhood.

### 2.3 Freeman encoding

The Chain Code algorithm by Freeman [1,3] is used to describe, using a string, the behavior of the contour of objects. This algorithm uses for orientation four possible directions (4-segments): North (2), South (6), Eastern (0) and West (4), or eight directions: North-East (1), North-West (3), South-East (7) and South-West (5). In fig.4(a) its presented the pattern for the four directions, used in the implementation of this technique.

The algorithm searches for objects in the image, covering all the borders and, in the end, returns to the start pixel. To elect the next marked pixel its analyzed and verified its neighbors. If its a boundary pixel, the current direction is upgraded to the elected direction of the next pixel until its found early. In fig.4(b) we show the directions proposed by Freeman and in fig.4(c), an example of a string sequence [7].

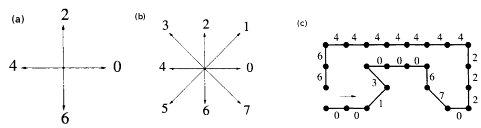

**Fig. 4.** (a) 4-directions of Chain Code. (b) 8-directions of Chain Code. (c) Example of Chain Code sequence.

### 2.4 Chain Code smoothing

The string produced by Chain Code algorithm generates redundant and unnecessary code to describe the contour, this way it is applied a second technique, which smooths this sequence [7] by analyzing every two characters, replacing, when necessary, with a character that describes the same behavior by using the 8-directions illustrated in fig.4(b), and is produced as a result a lower sequence. Some of these situations are illustrated in fig.5, where directional codes drawn in dotted form, are replaced by others drawn continuously. It is possible to see

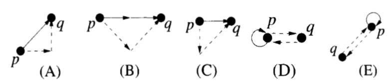

**Fig. 5.** Smoothing templates for Chain Code, adapted from [7].

that in fig.5(d) and (e), the pixel $q$ is removed, since it represents a "peak"in the outline.

### 2.5 Upper and lower profiles.

The characters sequence generated by the Chain Code algorithm represents the entire behavior of the objects contour, however, its necessary to represent the graphic behavior only the upper (maximum values) and lower pixel's profiles(minimum values), so,to find the upper profile, for each column of the image, the first white pixel(With a top-down/left-right scan) are organized into a string sequence and for each position of the sequence receive the row value of these white pixel. The process to find the lower profile are similar, the scan are made also for each column but starting with a bottom-up scan. To reduce noise and lost data we use the compute of average for each element in the string. Fig.6 show the upper(red line) and lower(blue line) profiles.

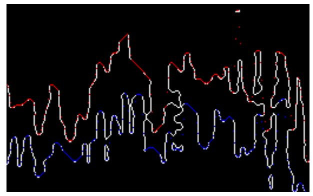

**Fig. 6.** Upper and lower profiles, highligthed in red(upper) and blue(lower). The white pixels are the Chain Code sequence detected for the object.

### 2.6 Points of maximum and minimum.

To detect the maximum and minimum points, we analyzed the concavity of the graphic by checking the inflection values for the upper and lower profiles. In this approach to detect points of maximum and minimum in the images, it was used the concept of vector. We tested different images, initially not using average filter, then with an average of 3, an average of 5, an average of 7 and an average of 9. The idea is based in a *Cartesian plane*, a vector $\boldsymbol{p} = \boldsymbol{AB} = \dot{B}$ - $\dot{A}$ and $\boldsymbol{q} = \boldsymbol{CD} = \dot{D}$ - $\dot{C}$, considering $\dot{A}$, $\dot{B}$, $\dot{C}$ e $\dot{D}$ are points in the graphic on the picture and $\dot{B} = \dot{C} = \dot{X}$, we have, if there is a transition from $\boldsymbol{p}$ to $\boldsymbol{q}$, such that, $\boldsymbol{p}$ is postive and $\boldsymbol{q}$ is negative or $\boldsymbol{q}$ is positive and $\boldsymbol{p}$ is negative, then the point $\dot{X}$ indicates a pixel that marks the maximum or minimum of a *f(x)* function of the graphic representing in the image. Follow is reproduced these algorithm.

**Initialize**:
- Set T as upper or lower Chain Code sequence;
- Set $i = 0$; Set $\{\alpha \mid \alpha$ is the vector magnitude, $\alpha \in \mathbb{N} - \{0\}\}$;

**Iteration**: $i$+$\alpha$...[*size*(T)-$\alpha$];
- *If* (((T[$i$]-T[$i$-$\alpha$] $>$ 0) AND (T[$i$+$\alpha$]-T[$i$] $<$ 0)) OR ((T[$i$]-T[$i$-$\alpha$] $<$ 0) AND (T[$i$+$\alpha$]-T[$i$] $>$ 0)));

  *If* (T[$i$] is the max/min value in the [i+$\alpha$ ... i-$\alpha$] array); Set T[$i$] = $\dot{X}$;

## 3 Experimental Results

Fig.7(b) is showed the result of applying the K-Means algorithm in fig.7(a) using Table 1(C3) values. It is possible to see that, the resulting image, which previously had no groupings, (i.e., there was no classification/*cluster* for the existing pixels), was divided into only three classes of color after running the algorithm. In this case, red for the graphic, green to the background and blue to the grid. This type of segmentation contrasts with Global Thresholding, which is based in only a single threshold T, as stated in [2], for the classification of objects in the image, insufficient for this image type, since its needed to segment three different objects, each with its respective threshold. The implementation of neighborhood removal is shown in fig.7(c). The colors used were arranged as follows: In red, the segmented graphic and green to the background.

As a result of the neighboring pixels analysis, the connection is made in some areas that were above separated in the resulting image of the K-means algorithm. The grid was previous removed by the neighborhood pixel analysis $p$. However, the graphic is also eroded in regions that have only grid as neighbor. This happens in regions where the trace made by the anemometer pen and grid intersect among themselves and the approximation performed by K-means doesnt segment appropriately, as shown in fig.7(b) and (c). Using average filter, the contours of objects are smoothed and some disconnected regions are regrouped [2,4], as shown in fig.7(d).

In fig.7(e), it is possible to analyze the result of the Chain Code algorithm [1] to the edge of the chart speed, applied to the image smoothed by the filter average. With this, we are able to describe object boundary with only one sequence of characters, that will be used later to analyze the behavior of the graphic. The smoothing of the Chain Code reduces the imperfections and redundancy [7] of the boundary found, this is important for the quality of maximum and minimum points analysis, that should be detected in the graphic, since it reduces the number of pixels marked incorrectly. In fig.7(f), the graphic is represented the upper

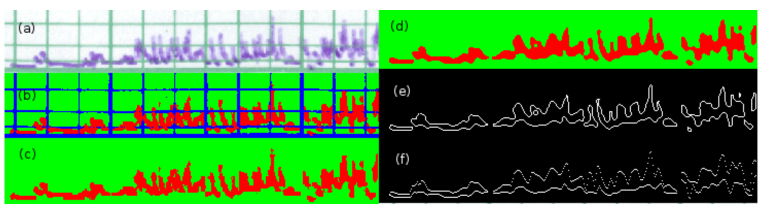


**Fig. 7.** (a) Input image. (b) K-Means segmentation. (c) Removal of neighbor. (d) Average filtering. (e) Chain Code sequence. (f) Upper and lower profiles with average 3.

and lower profiles using average 3; For the experiments, we used fig.1. The fig.8, showed the best results with the filter average 3 in the boundary smoothing. The pixels marked with black in upper and lower profile indicates points of maximum and minimum. It is possible to see there is a distance, on some pixels between the right positions of the of maximum/minimum points,(i.e., drawn by the pen of the anemometer), and those detected by the software. The correct position of the points in the image are highlighted with small $x$ letter, by the author.

## 4 Conclusion and future work

It was observed by analyzing the results shown in fig.8, the pixels that characterize the points of maximum and minimum were mostly detected with considerable accuracy. However, despite the satisfactory results, some of these pixels have a

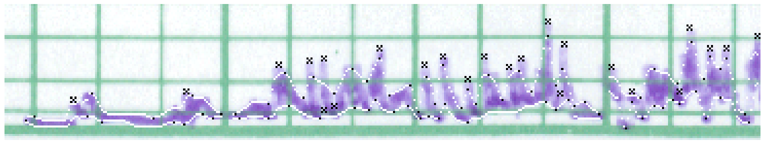

**Fig. 8.** Parcial result image. Black pixel's indicates inflexion(maximum/minimum) points detected. Letter $x$ indicates the right position.

small shifting when compared to the graphic on the image. This undesirable behavior occurs for several reasons, including imperfect approximation of *clusters* by K-Means algorithm; information loss caused by the average filtering; information loss caused by noise filtering of the Chain Code sequence and upper and lower profiles sensitive to it. Futher research must be conduct to reduce these situations and deal with low quality or damaged images. The next steps included too the vetorization of this segmented image and subsequent storage in a appropriated database, to help the search and recovery process. The algorithms are implemented using the computer vision library OpenCV[6].

## 5 Acknowledgments

We would like to thank Jander Moreira for the help and orientation on this work.

## References


1. H. Freeman: On the Encoding of Arbitrary Geometric Configurations. Electronic Computer (1961)
2. Gonzalez R. C., Woods R. E.: Digital image processing. Addison-Wesley (1992)
3. Nakashima K., Koga M., Marukawa K., et al.: A High-Speed Character Contour-Fill Method Using an Edge-Flag List. IAPR Workshop on Machine Vision Applications (1994)
4. Castleman K. R.: Digital Image Processing. Prentice Hall (1996)
5. Wenyin L.,Dori D.: A Generic Integrated Line Detection Algorithm and Its Object-Process Specification. Computer Vision and Image Understandig 420-437 (1998)
6. Intel: Open Source Computer Vision Library. (2000)
7. Marchard-Mailet S., Y.M. Sharaih: Binary Digital Image Processing A Discrete Aproach. Academic Pres (2000)
8. Allaby M.: Encyclopedia of weather and climate. Infobase Publishing (2002)
9. A. Takahashi and B.R.C. Bedregal and A. Lyra: Uma Versão Intervalar do Método de Segmentação de Imagens Utilizando o K-Médias TEMA Tend. Mat. Apl. Comput (2005)
10. Forsyth D. A. and Ponce J.: Computer Vision: A Modern Approach. Prentice Hall (2002)