Image Pattern Recognition in Weather Forecast

Luís Emílio Cavechiolli Dalla Valle
Ricardo Luis de Azevedo da Rocha, *Member, ACM, SBC*

***Abstract*—As a parallel way to study the weather, a new model of pattern recognition in satellite image sequences is addressed. A grammar for a language the symbols of which are based on some image sequences was created; their derivation rules are generated by counting the number of times a symbol is found in an image, and the number of times it changes to another symbol in the next image.**

I. Introduction

The analysis of a weather satellite image sequence, ordered in time, shows very complex vapor steams motions that travel the atmosphere of the planet. The seemingly chaotic movements of this events hinder, at the first moment, the achievement of patterns or repetitions(recurrences), which allows forecast or computational inferences. Due to that, it is necessary to simplify the image, to the point that it turns most easily interpreted and light as possible, loosing the minimum important data along this work. To achieve this relative simplicity (reduce the problem domain complexity [1]), given that the initial goal is to model the steam movement, the original METEOSAT9 satellite image provided by Instituto Nacional de Pesquisas Espaciais (INPE)[2] (figure 1) was transformed into a black and white image.

The filtering process was modeled by applying the median filter to remove the states boundaries of the analyzed region, and threshold to remove the remaining colors, which are not clouds, also normalizing it, to transform the cloud steams into a regular white layer, to contrast with the rest that is not a cloud, as illustrated in figure 2.

With the simplified version of the image, the Canny's edge detection algorithm, described in [3] was used, based on the [4] implementation, generating an image sequence, showing just the clouds edges. The example of the result is shown in figure 3.

The Canny's edge detection algorithm [3] guarantee the one pixel edge thickness required to implement the proposed language.

II. Motivation

The use of a less expensive way to forecast weather and, at the same time, present good outputs (even when compared to those obtained from super-computerized numerical models) is the main motivation. The questions underneath this research are:

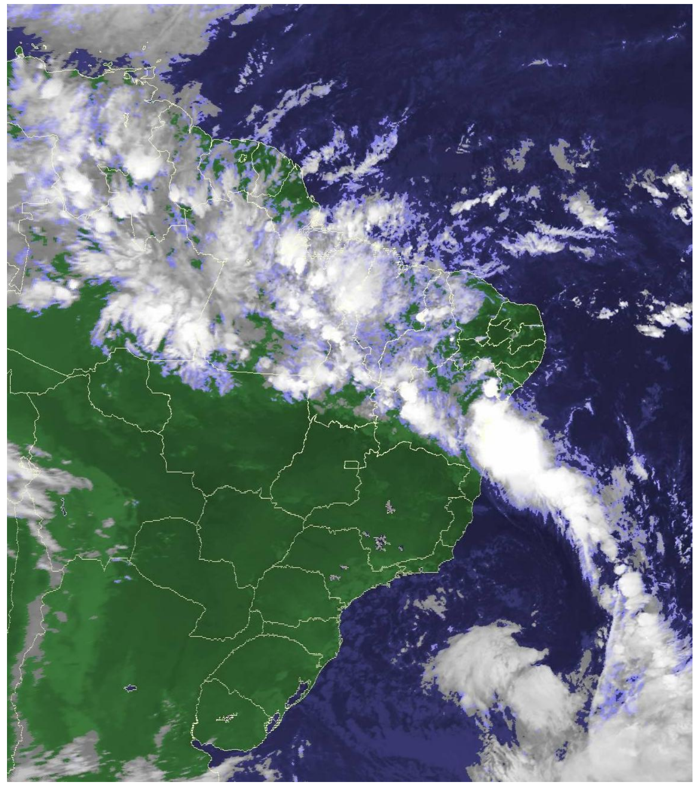


Fig. 1: Original Image. Source: INPE

- Is it possible to use just a satellite image sequence to predict the weather behavior (forecasting), similar to the actual super-computerized numerical models?
- Is it possible to infer some data, such as atmospheric pressure, temperature, humidity, by viewing only these images?

III. The Symbols

The image information unit is the pixel; however, for a pattern movement analysis, a pixel is not enough information. The most convenient is a 3×3 matrix, because it contains the central pixel, and at least one neighbor in each direction. Thus it is possible to fix a central point of interest (in a central pixel) and understand the general movement of that region, analyzing the neighborhood. As a consequence, the manipulated image that represents the symbols has three pixels in width by three pixels in height, displayed as follows:

$$M = \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{bmatrix}$$

where p_{22} is the reference central pixel.

Thanks to CNPq for supporting Luís Emílio

L. E. C. Dalla Valle: Graduate Student at Engineering School of University of Sao Paulo. E-mail: luisdallavalle@usp.br

R. L. A. Rocha: Laboratory of Languages and Adaptive Technology, Engineering School of the University of Sao Paulo. E-mail: rlarocha@usp.br

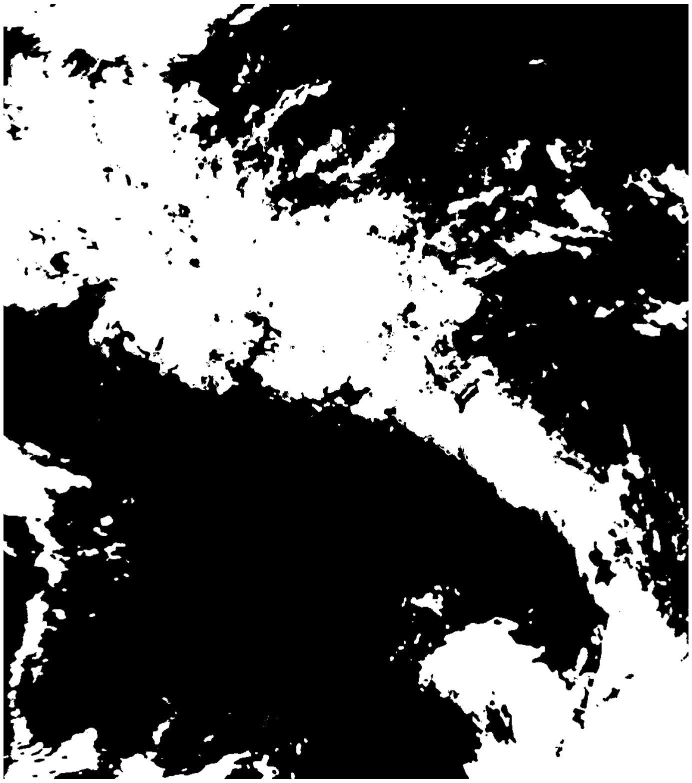


Fig. 2: Image in B&W

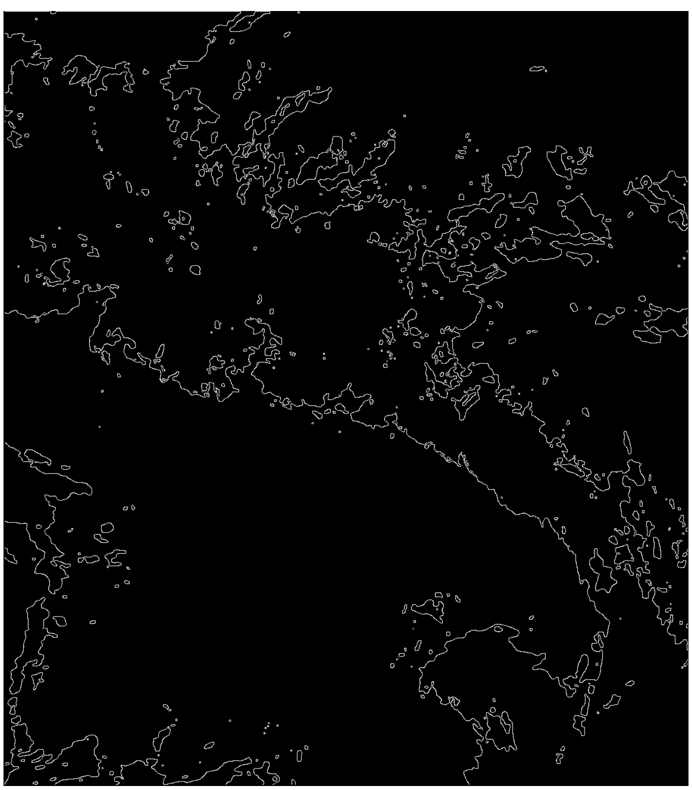
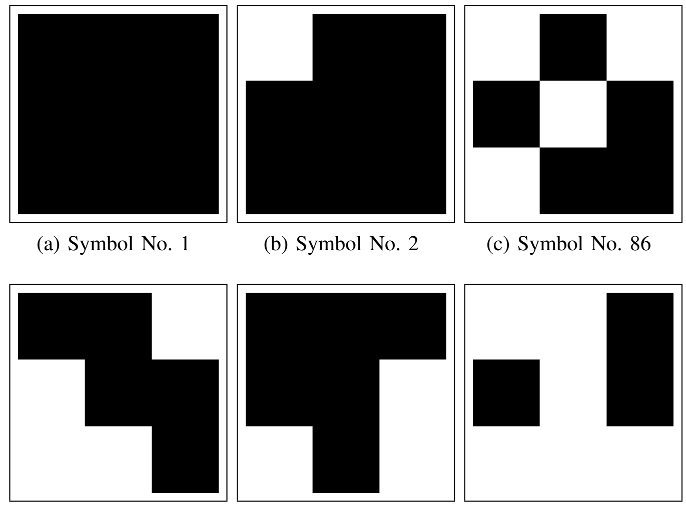


Fig. 3: Edge Detection using Canny

The matrix above represents a binary image $B : Z^2 \rightarrow \{0, 1\}$, that contains nine pixels, and can be represented by a binary chain of the form $p_{33}p_{32}p_{31}p_{23}p_{22}p_{21}p_{13}p_{12}p_{11}$, with 2^9 different combinations of zeros and ones, summarizing 512 images, where 0 is black and 1 is white.

Figure 4 shows some examples of the generated symbols for the language.



(a) Symbol No. 1 (b) Symbol No. 2 (c) Symbol No. 86

(d) Symbol No. 205 (e) Symbol No. 353 (f) Symbol No. 468

Fig. 4: Example of symbols created by the combination of black and white pixels in a 3×3 matrix

IV. Analysis

Assuming that the image edges have specifically one pixel in thickness, granted by the Canny algorithm, a *match* function discovers which symbol corresponds to which image part, both in the current and the next image in the sequence, and fills each image part with that symbol found. The procedure is repeated for all the images in the sequence.

When this process finishes, the number of transitions are count (from one symbol to another). For example, if in the first position of the first image, symbol No. 1 was matched, and in the next image, in the same position, symbol No. 2 was matched, value 1 is added to the No. 1 to No. 2 transition counter. This process is repeated for each superposed symbol in the images.

The process is repeated for eight more times, and for each new repetition, it shifts one pixel to the right or one pixel downwards, to avoid non-examination of any pixel.

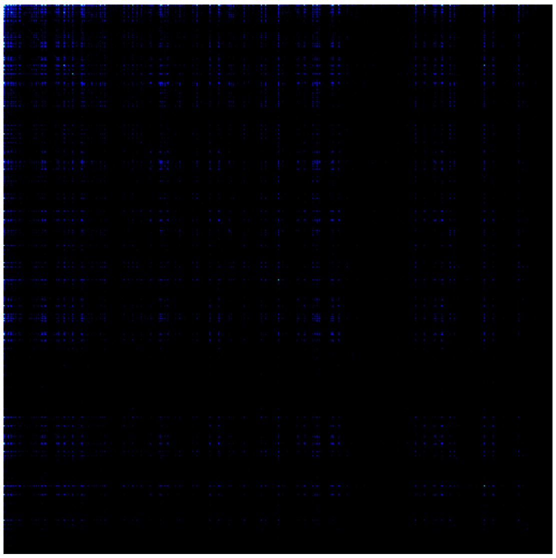
Both the symbols occurrences and derivation [5][6] counters, are stored in a database, and, for optimization purposes, only the occurred transitions are stored, implying the zero valued transitions. Otherwise there would be $\frac{n!}{(n-2)!}$, for $n = 512 \Rightarrow 261.632$ stored transitions.

To represent the transitions density, the results were shown in a 512 pixels wide by 512 pixels high map, where each line represents the initial image symbols, the columns represent the transitioned symbols, and the pixel color in a given coordinate represents the transition counter value (from line symbol to column symbol).

Figure 5a shows the density transitions map for the minimum possible data, just two images in the image sequence. Subsequent figures contain a greater transition accumulation, because the analysis was repeated for a greater amount of images in the sequence, 10, 30 and 60, the mapping of which is shown in figures 5b, 6a e 6b respectively[1]

Fig. 5: First two tests

(a) Result of *Match* for 2 accumulated images



(b) Result of *Match* for 10 images

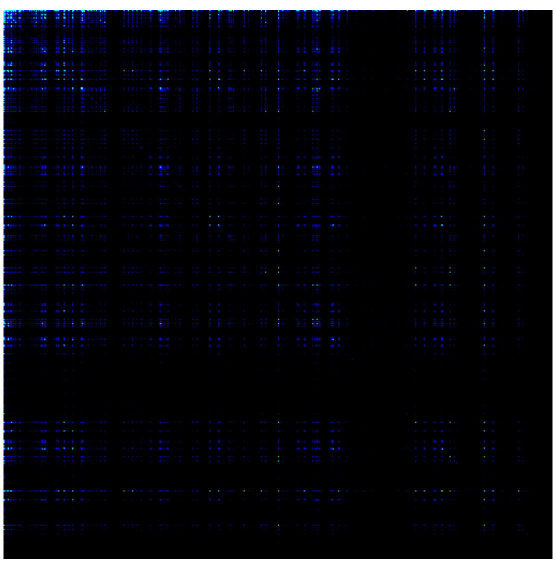
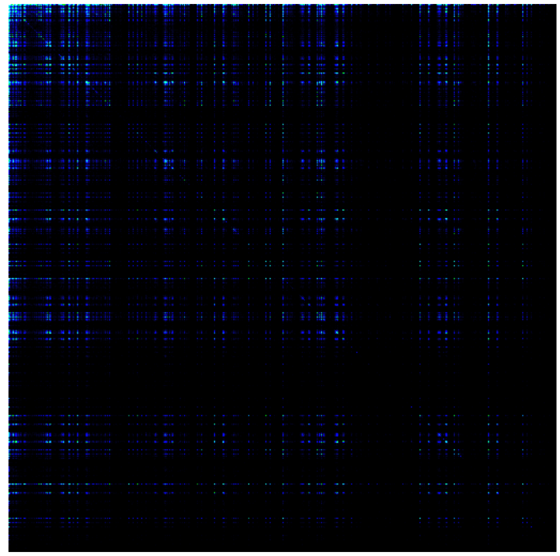
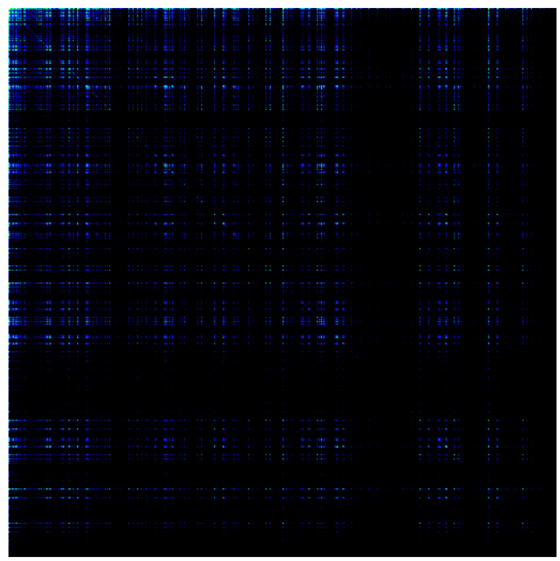


Fig. 6: Last two tests

(a) Result of *Match* for 30 images



(b) Result of *Match* for 60 images



The first lines and columns have a great number of transitions, because most of the image is composed of black, corresponding to symbol No. 1, which implies in future removal in further studies, because only the significant symbols in resonance with detected edges will be maintained.

A diagonal component was verified, from pixel $(0, 0)$ to $(512, 512)$, which indicates the absence of transition. This occurs due to the slow movements of some analyzed edges, to casual correspondences from the same symbol (randomly), or to atmosphere circulatory movements. This diagonal component will tend to become less evident with a better model calibration.

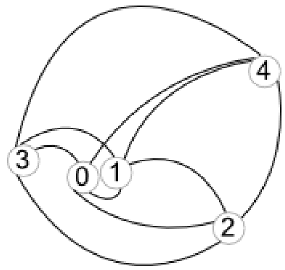
It worth noting that there is a dissipation from the first symbols to the last ones. Apparently, because the edges have better correspondence in the first ones, which have more black pixels then white, forming linear patterns than the last ones, which form white dense conglomerates.

Finally, a graph is constructed with symbols as it's vertex, and the transition count as edges, as illustrated in Figures 7a and 7b where the edge size is inversely proportional to the counter (weight) which has the transition, to make the heavier vertex closer and set aside the lighter vertex, to get another visual answer to this process.

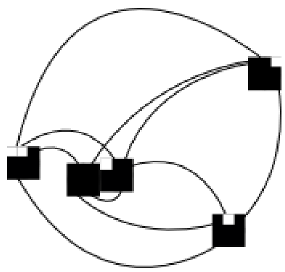
[1]For typographic reasons, these images were modified in contrast and saturation to highlight the transition occurrences

Fig. 7: Graph of the first symbols

(a) Symbol id as vertex



(b) Symbol as vertex



V. Conclusion

This research is a work in progress, as there are problems to solve, and also much error propagation in the process over the analysis time, such as:

- Image Quality - The image acquisition is not 100% true, and the filters increase this inaccuracy.
- Image Size - the approach of this work is strongly image resolution dependent. It needs to be fixed, and as large as possible, since for each resolution, a different pattern will be matched, generating different data sets.
- Satellite Image Acquisition Interval - For a more consistent analysis, the time between the first image and the subsequent image must guarantee that the maximum distance that an edge moves is two pixels, to stay within the limits of the created language. Otherwise the symbols must have different sizes, such as 5×5, 7×7, etc, all odds, to maintain the central pixel as a reference mark. Yet doing this, the 512 symbols would be 33,544,432 and 562,949,953,421,312 respectively, oversizing the problem.
- Image Filtering - When applied, some filters such as *Threshold* and edge detection can cause some loss of information about the original shape of vapor steams.
- Pattern Matching - With the above items, the patterns matched in images may not correspond exactly to the true ones, despite being close, generating incorrect derivation weights.

When starting a new image processing techniques study during this work development, new possibilities for image processing were discovered, such as to almost extinguish the loss of quality and error propagation, in filtering application, also creating interpolated images, to smooth the image sequence, allowing a more precise symbol transition, generating a reliable map for further analysis. However, this improvements will be implemented in a next step.

A deeper study about the actual meteorology and climatology is necessary, to better lapidate the proposed idea and verify what can be said about climate behavior and grammar formation rules, if they exists, and how the relationship with this proposed work approaches.

With the above upgrade conclusion, adaptive devices [7][8][9] [10] will be used in the analysis result based on the inverse way. Providing an initial symbol set, to form a very first image, the adaptive device will reconstruct the next images, based on formation rules formed from initial analysis, and find patterns in a greater results set, for example, comparing maps by seasons, or any other relevant recurrent climate event.

With this, searching for a less expensive form of weather forecast.

References

[1] C. H. Papadimitriou and H. R. Lewis, *Elements of the Theory of Computation*. Prentice-Hall, 1998.

[2] CPTEC, "Centro de previsão do tempo e estudos climáticos do instituto nacional de pesquisas espaciiais," june 2010, acessado em junho/2010. [Online]. Available: http://www.cptec.inpe.br/

[3] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, pp. 679 – 698, 1986.

[4] M. Heath, S. Sarkar, T. Sanocki, and K. Bowyer, "A robust visual method for assessing the relative performance of edge-detection algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 1338–1359, 1997.

[5] F. Baader and T. Nipkow, *Term rewriting and all that.* Cambridge University Press, 1999, ch. 1, p. 316.

[6] N. Dershowitz and J.-P. Jouannaud, *Rewrite Systems*. Elsevier and MIT Press, 1990.

[7] J. J. Neto, "Contribuição à metodologia de construção de compiladores," Ph.D. dissertation, Universidade de São Paulo, 1993.

[8] J. José Neto, "Adaptive rule-driven devices - general formulation and case study," *Lecture Notes in Computer Science*, vol. 2494/2002, pp. 466–470, 2002.

[9] J. J. Neto and C. Bravo, "Adaptive automata - a reduced complexity proposal," *Lecture Notes in Computer Science*, vol. 2608/2003, pp. 1–46, 2003.

[10] H. Pistori and J. J. Neto, "Adaptree - proposta de um algoritmo para indução de Árvores de decisão baseado em técnicas adaptativas," *Anais Conferência Latino Americana de Informática*, vol. Novembro/2002, pp. 1–10, 2002.

Luís Emílio Cavechiolli Dalla Valle was born in São Paulo / SP on May, 18th, 1980. With a degree in Computer Engineering, from Faculdade de Engenharia of Fundação Santo André - FAENG/FSA, in 2008, he is currently studying for his MSc at the Engineering School of the University of São Paulo.

Ricardo Luis de Azevedo da Rocha was born in Rio de Janeiro / RJ on May, 29th, 1960. He was granted a degree in Electronic Engineering, focus on Digital and Computer Systems from Pontifícia Universidade Católica do Rio de Janeiro - PUC-RJ, in 1982, received his MSc (1995) and PhD (2000) in Computing Engineering from the Engineering School of the University of São Paulo - EPUSP. Currently he is a professor at the Engineering School, and his research interests are Adaptive Techniques and Technologies, Computational Models, Intelligent Systems, Kolmogorov Complexity and Foundations of the Theory of Computation. Dr. Rocha has been a member of the Association for Computing Machinery ACM since 2001, and also of the Brazilian Computing Society - SBC since 2000.