Image Filtering by Using Specific Hard Thresholds on Single Values

Nilton Correia da Silva, MSTM/ UniEVANGÉLICA
Antonio Nuno de Castro Santa Rosa, IG/UnB

Copyright 2010, SBGf - Sociedade Brasileira de Geofísica

Este texto foi preparado para a apresentação no IV Simpósio Brasileiro de Geofísica, Brasília, 14 a 17 de novembro de 2010. Seu conteúdo foi revisado pelo Comitê Técnico do IV SimBGf, mas não necessariamente representa a opinião da SBGf ou de seus associados. É proibida a reprodução total ou parcial deste material para propósitos comerciais sem prévia autorização da SBGf.

Abstract

This work explores the spatial relation of Single Values over image pixels in order to propose a filtering process. Different denoise degrees are applied to each pixel considering their effects on a non supervised clustering process – Self Organizing Map. This results on specific filtering processes to different spectral characteristics of the images. Two experiments are presented; the first one with synthetic data and the second with LandSat-7 data. The first one considers a scene with high frequencies and consequently cuts only the null space of the single values of each pixel. The experiment with LandSat-7 data shows a case with homogeneous scenes. In this case, the filtering process implements hard cuts considering a limited group of classes. The technique presented here brings a finely way to reconstruct better approximations of the original data, and at the same time, excludes unnecessary ranges of pixel variations.

Keywords. Single Values Decomposition, Self-Organizing Maps, Filtering, Image Processing.

Introduction

It is difficult to apply a hard threshold in the Single Values which reduces the noise in images and, at the same time, doesn't make the filtered image loses important features which characterize objects in the real scene. To avoid this problem the authors have studied the contributions of the single values on the reconstructed pixels. Choosing the same single value subset to reconstruct all pixels result on different effects in different regions of the image, because one single value has different contribution degrees on different pixels. A maximum cut on the reconstructed pixel values is achieved by using a minimum quantity of single values on its reconstruction. In this work we use a non supervised neural network – Self Organizing Map – to regulate the minimum quantity of single values to each pixel of the reconstructed image. Self-Organizing Map (SOM) (section 3) is an algorithm which has the special property of creating spatial organized representations. It is very useful as an unsupervised clustering method [6]. Its spatial representations are used, in this work, to regulate the sizes of the single values subsets, which are used to reconstruct the filtered image.

SVD and Hard Threshold

The equation 1 brings the inverse process of a Single Values Decomposition (SVD) [1] applied on the image $\overline{\overline{D}}$. This decomposition is known as Single Values Decomposition because the values of the matrix $\overline{\overline{S}}$ are the positive square roots of the eigen values of $\overline{\overline{D}}^t.\overline{\overline{D}}$ or $\overline{\overline{D}}.\overline{\overline{D}}^t$. So $\overline{\overline{S}}$ is designed single values of $\overline{\overline{D}}$ [1].

The diagonal matrix $\overline{\overline{S}}$ is sorted by contribution degree of its single values on the reconstruction process.

$$\overline{\overline{D}}' = \overline{\overline{U}} \cdot \overline{\overline{S}} \cdot \overline{\overline{V}}^t \qquad (1)$$

By the analysis of the diagonal matrix $\overline{\overline{S}}$, it is possible to verify the contribution degree of its individual values in different regions of the reconstructed image D'. If it is defined an index $0 < k < N$ and changes all value of $\overline{\overline{S}}$ with indexes greater than k to zero, the reconstructed image will be a softened image (2). The changes in the pixels will depend of how much *k* is close of one.

$$\overline{\overline{D}}' = \overline{\overline{U}} \cdot \begin{bmatrix} S_{11} & 0 & 0 & 0 & . & 0 \\ 0 & . & . & . & . & 0 \\ 0 & . & S_{kk} & . & . & 0 \\ 0 & . & 0 & 0 & . & 0 \\ . & . & . & . & . & . \\ 0 & 0 & 0 & 0 & . & 0 \end{bmatrix} \cdot \overline{\overline{V}}^t \qquad (2)$$

This filtering process, by using a hard threshold defined by the constant *k*, results on a softened image which minimize the noise effects presented in the high frequencies of the original image [2,3]. But applying the same cut criterion to all pixels means to have different effects over different regions of the reconstructed image. In section 4 we propose a filtering process which uses particular constants *k* to each reconstructed pixel.

Self Organizing Map (SOM)

An important feature of neural networks is the ability to *learn* from their environment, and through learning to *improve* performance in some sense. They are classified in supervised and unsupervised learning methods. In the supervised method the targets may take the form of a desired input-output mapping that the algorithm is required to approximate. In another way, the purpose of a self-organizing map is to *discover* significant patterns or features in the input data *without a teacher*. The algorithm is provided with a set of rules of *local* nature which enables it to compute an input-output mapping with specific desirable properties; the term "local" means that the change applied to the synaptic weight of a neuron is confined to the immediate neighborhood of that neuron (figure 1) [4,5].

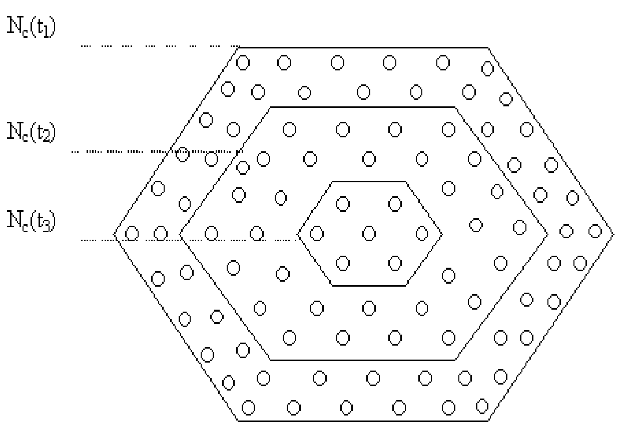


Figure 1. Output layer and the neighborhood function: $N_c(t)\{t_1 < t_2 < t_3\}$.

The learning process of a self organizing system consists of repeatedly modifications in the synaptic weights of all connections in the system in response to input (activation) patterns and in accordance with prescribed rules, until a final configuration develops [7]. Initially, consider one matrix of neuron units, one weight vector $m_i(t) \in R^n \{t = 1,2,3,...\}$ associated to each unit i and one input pattern $x(t) \in R^n$ being compared with all units. The following two rules explain how the map is generated by self organization [9]:

Rule 1: To find the unit c which weight vector is more similar to the input pattern $x(t)$:

$$\|x(t) - m_c(t)\| = \min_i \{\|x(t) - m_i(t)\|\} \quad (3)$$

The unit c is considered the unit which responds to the pattern $x(t)$.

Rule 2: To modify the weight vectors of the unit c and its topologic neighbors:

$$\left.\begin{array}{l} m_i(t+1) = m_i(t) + \alpha(t)[x(t) - m_i(t)] \forall i \in N_c \\ m_i(t+1) = m_i(t) \forall i \notin N_c \end{array}\right\} \quad (4)$$

The topologic neighborhood N_c is a time dependent function (figure 1). $\alpha(t)$ is the learning rate of the winner neuron $(0 < \alpha(t) < 1)$. This rate and the N_c diameter decrease during the learning time [9].

In all experiments presented below, a one-dimensional matrix was used with 50 neuron units, $\alpha(0)$ = 0.8 and the $N_c(0)$ diameter was maximum – covering all neurons (in this case, the neighborhood topology is linear) [8,9]. The input patterns are the pixel amplitude values – codified with 8 bits (values from 0 to 255). At the end of 2000 learning epochs [8,9], the resultant maps are used as clusters sets in the algorithm of the *k* constants matrix calculation.

***K* constants matrix calculation**

In order to improve the filtering process that uses a unique threshold to all original pixels (see section 2), we propose to generate specific points of hard threshold to each pixel of the original image.

To avoid the reconstructed image (filtered image) loses the target definitions presented in the original image scene, all filtered pixels must belong to the same original pixel cluster. In this way, the first step is to generate the cluster set from the original image (using SOM, see section 3).

Its necessary to create a cluster map (with the same image dimensions) which has the respectively cluster of each original pixel. This is done by using the equation (3) to each pixel and writing the cluster index in the respectively cluster map cell (figure 4). This cluster map will be used to control the quantity of single values that will be used to reconstruct each filtered pixel. One filtered pixel (or a pixel reconstructed with a minimum quantity of eigen values) must be classified by the same cluster of its respectively original pixel. Then, the filtering algorithm needs to define a matrix $\overline{\overline{K}}$ of constants (one constant to each pixel of the original image). At the last step, the pixels of the filtered image $\overline{\overline{F}}$ are calculated by equation (5):

$$F(i,j) = \overline{U_i} \cdot \begin{array}{|c|c|c|c|c|c|} \hline S_{11} & 0 & 0 & 0 & . & 0 \\ \hline 0 & . & . & . & . & 0 \\ \hline 0 & . & S_{k(i,j)} & . & . & 0 \\ \hline 0 & . & 0 & 0 & . & 0 \\ \hline . & . & . & . & . & . \\ \hline 0 & 0 & 0 & 0 & . & 0 \\ \hline \end{array} \cdot \overline{V_j}^{\,t} \quad (5)$$

Experiment with synthetic data

The filtering process was applied in a synthetic image (100x100 pixels) with 4 different gray degrees (figure 2) in order to illustrate the filtering algorithm.

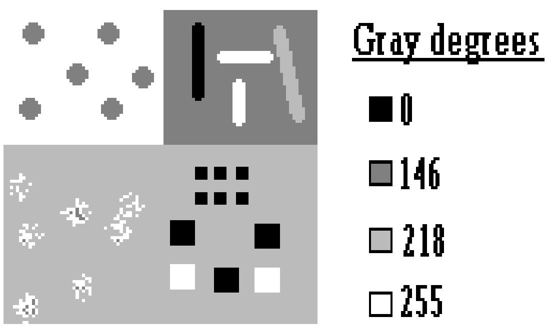


Figure 2. Synthetic image.

In this case, the 1st, 16th, 31st and 50th SOM clusters classified the four different pixel values present in the synthetic image (figure 3).

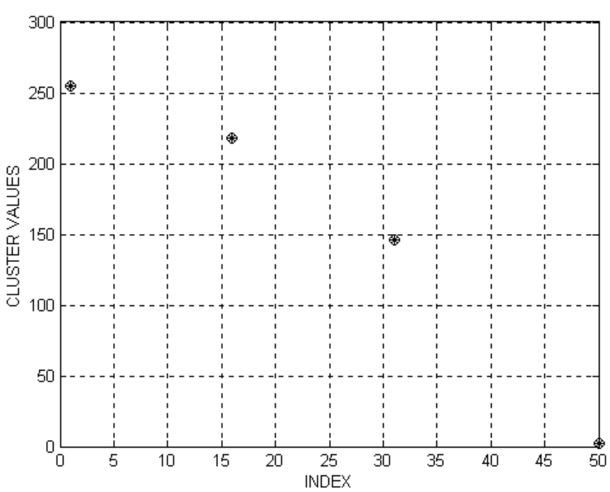


Figure 3. Clusters detected in the synthetic image

The figure 4 shows the clusters spatial localization. This cluster map was generated by classifying the original image (figure 2) with the four clusters detected by the SOM learning algorithm (figure 3).

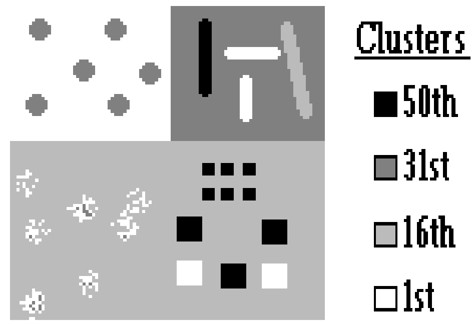
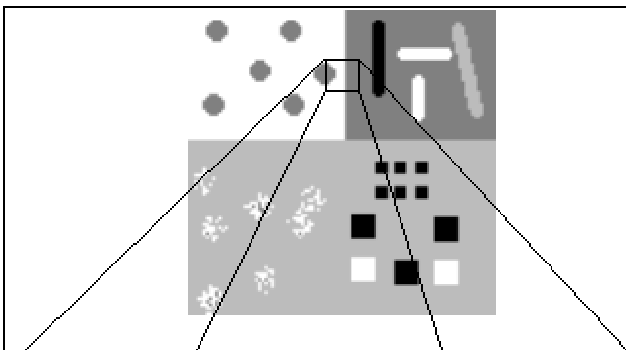


Figure 4. Cluster map of the Synthetic image.

The figure 5 shows a projection of part of the $\overline{\overline{K}}$ matrix calculated in this application. The minimum and maximum quantities of eigen values used to reconstruct the filtered image was 3 and 60 respectively.



059	060	059	051	048	048	048	051	051	040	040
026	057	046	059	057	057	057	051	051	040	040
030	057	046	059	057	057	057	051	051	040	040
060	059	059	060	060	060	060	057	057	042	042
057	057	059	057	057	057	057	052	052	040	040
057	058	057	058	048	048	048	052	052	047	047
055	055	054	055	057	057	057	050	050	039	039
055	055	054	055	057	057	057	050	050	039	039
055	057	055	055	048	048	048	052	052	047	047
057	057	057	060	058	058	058	053	053	047	047
057	058	057	060	060	060	060	055	055	040	040

Figure 5. *K* constants used to filter the pixels of the region limited by the black square.

The figure 6 shows the whole $\overline{\overline{K}}$ matrix in gray scale (the values vary from 3 to 60). It's easy to see the high frequency regions are composed by more eigen values linear combination and homogeneous regions are reconstructed by one or few eigen values combinations.

The pixels of the filtered image (figure 7) were calculated by using the equation 5 with the $\overline{\overline{K}}$ matrix of the figure 6.

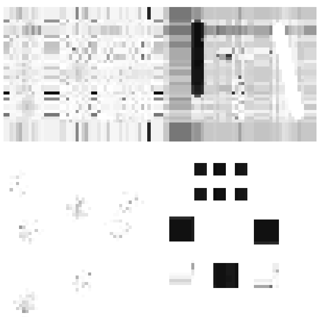


Figure 6. K matrix in gray scale.

The figure 8 shows the residual between the original synthetic image ($\overline{\overline{D}}$) and the filtered image ($\overline{\overline{F}}$). The minimum and maximum values of this residual matrix are 0 and 20 respectively.

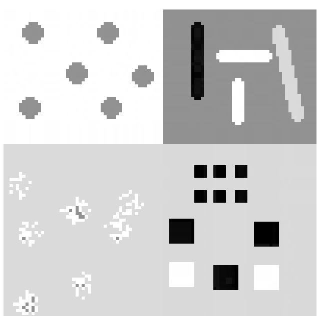


Figure 7. Filtered image.

The residuals shown in the figure 8 demonstrate the most affected regions by the filtering process.

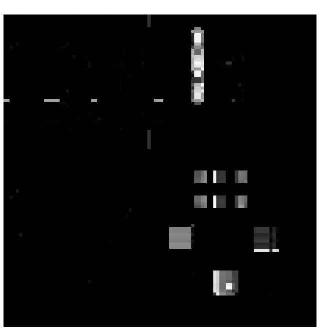


Figure 8. Residual matrix.

Experiment with LandSat-7 data

The filtering process was applied in a 8 bits LandSat-7 image, band 4, time: 07/20/2001 – 13:02:34.000, from Alto Paraíso city / Goiás state / Brazil (Figure 9).

Figure 9. Original Image (200x200 pixels).

The weights of the SOM net (representing the 50 clusters) generated from the pixel values present in the original image can be seen in the figure 10.

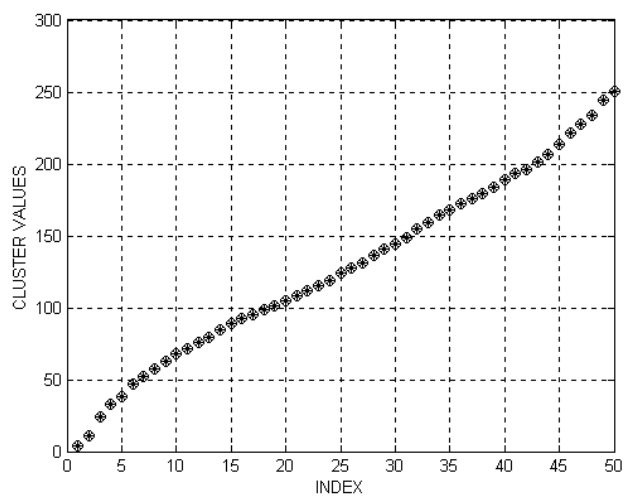
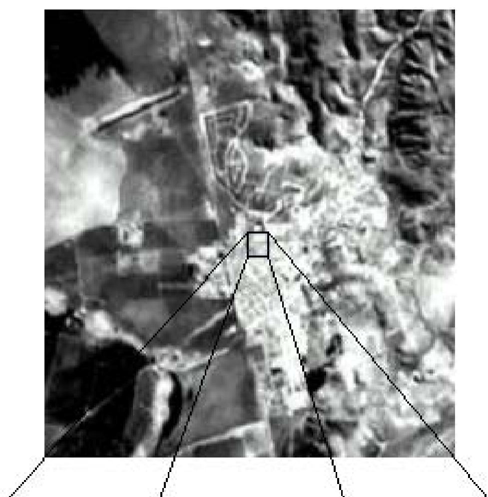


Figure 10. Clusters detected in the original image.

The figure 11 shows the clusters spatial localization. This cluster map was generated by classifying the original image (figure 9) with the 50 clusters generated by the SOM learning algorithm (figure 10).

Figure 11. Cluster map of the LandSat-7 image.

The figure 12 shows a projection of part of the $\overline{\overline{K}}$ matrix calculated in this application. The minimum and maximum quantities of eigen values used to reconstruct the filtered image was 2 and 198 respectively.



112	083	119	069	140	129	106	080	103	062	086
109	090	164	113	021	109	102	084	092	091	016
100	094	073	065	068	054	074	128	109	142	090
112	127	162	147	174	100	123	133	098	148	071
145	096	080	165	077	169	130	131	111	164	091
112	120	108	100	111	113	084	067	083	097	112
122	070	110	079	093	068	101	067	065	062	065
142	106	078	155	174	106	082	082	072	089	082
140	141	119	164	155	088	100	085	106	079	076
074	094	138	082	061	106	117	080	069	047	074
145	145	074	059	083	095	055	145	117	079	113

Figure 12. *K* constants.

The figure 13 shows the whole $\overline{\overline{K}}$ matrix in gray scale. As in the synthetic case, the high frequency regions are composed by more eigen values linear combination and homogeneous regions are reconstructed by one or few eigen values combinations.

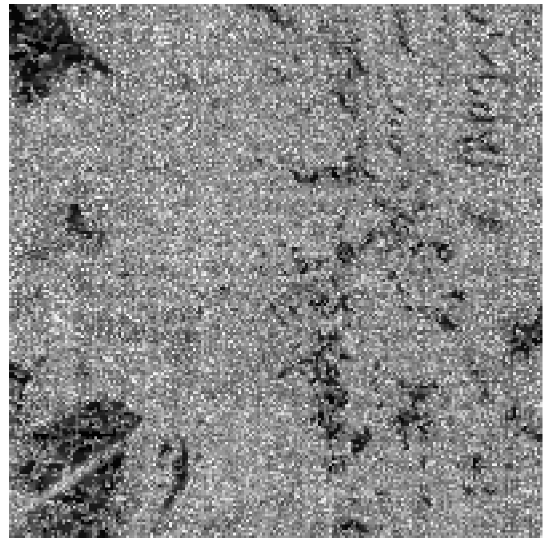


Figure 13. *K* matrix in gray scale.

The figure 15 shows the residuals between the original synthetic image ($\overline{\overline{D}}$) and the filtered image ($\overline{\overline{F}}$) see figure 14.

Figure 14. Filtered image.

The minimum and maximum values of the residual matrix are 0 and 39 respectively.

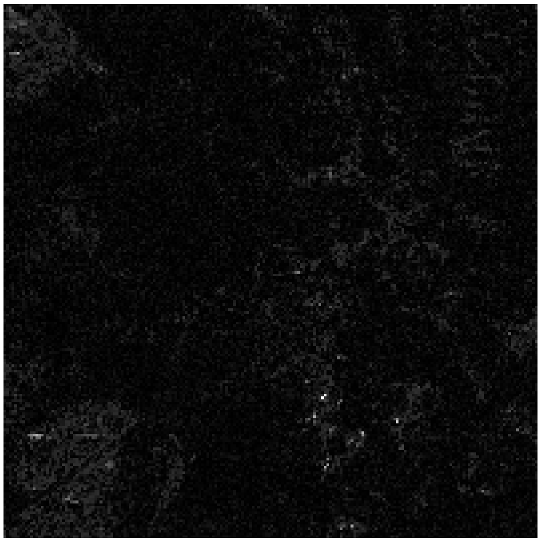


Figure 15. Residual matrix.

The residuals shown in the figure 15 demonstrate the most affected regions by the filtering process.

Conclusions

The Self Organizing Map was important to define an independent filtering process. Due its unsupervised classifier feature, there wasn't interference in the process of generate the clusters set and consequently the *K* matrix.

The final result of the filtering can be tuned by SOM training algorithm parameters. In the case of the synthetic data, the quantity of clusters was super estimated, generating one cluster to each pixel sample. In this case, the thresholds (*k* values) were very precise and consequently cut just a small part of the original values – basically, this experiment cut the null space of the $\bar{\bar{S}}$ matrix. This parameterization case works in sensible data, where details of the image (high frequency) must be preserved.

When homogeneous targets visualization must be improved, the under estimation of the SOM size is recommended. In the case of the LandSat-7 data experiment, 256 pixel samples were grouped in 50 clusters. The figure 15 shows the response to this choice – the residual matrix shows cuts from 0 until 39 in the original pixel values. Different results could be achieved by using other SOM geometries: smaller maps will result in greater cuts.

Changes in the single values matrix could have many effects in the reconstructed image. This technique arise in a finely way to reconstruct better approximations of the original data, and at the same time, excluding unnecessary ranges of pixel variations.

Acknowledgments

Dr. Nilton Correia da Silva was partially supported by a CAPES scholarship.
The LandSat-7 data was supplied by INPE.

References

[1] B. Noble, J. W. Daniel, “Álgebra linear aplicada”, PHD, 2ª Edition (1986), 262--280.

[2] Medeiros F. N. S, Mascarenhas, N.D.A., Costa, L. F. , Adaptive Speckle MAP Filtering for SAR Images Using Statistical Clustering. Anais SIBGRAPI, **V.** I , pp. 303 – 310, October 1998.

[3] Sant’anna S. J. S.; Mascarenhas N. D. A. , Avaliação comparativa da perda de resolução espacial de filtros redutores de ruído *Speckle.* Anais SIBGRAPI, **V.** I , October, 1994.

[4]Ritter H.; Schulten K.. "Convergence properties of Kohonen's topology conserving maps: Fluctuations, stability and dimension selection". *Biol. Cybernetics*, 60, pp. 59--71, 1988.

[5] Malsburg C. von der, "Network self-organization", In: An introduction to neural and electronic networks (S.F. Zornetzer, J.L. Davis, C. Lau, eds.), pp. 421-432. San Diego, CA: Academic Press, 1990.

[6] Weigang, L; Silva, N. C, “Implementation of Parallel Self-Organizing Map to the Classification of image”, SPIE’99, **V**. 1, (1999), 284--292.

[7] S. Haykin, “Neural Networks – A comprehensive foundation”, *Macmillan College Publishing Company,* (1994), 352-354.

[8] J. E, Darhoff, "Neural network architectures: an introduction", Van Nostrand Reinhold, (1990).

[9] T., Kohonen. "Speech recognition based on topology-preserving neural maps ", (1990). “Neural Computings Architectures: The Design of Brain-Like Machines” - Aleksander, I., pp. 26-40. MIT Press, Cambridge, Massachusetts, USA.