# Non-supervised Classification of 2D Color Images Using Kohonen Networks and a Novel Metric

Ricardo Pérez-Aguila, Pilar Gómez-Gil, and Antonio Aguilera

Departamento de Ingeniería en Sistemas Computacionales,
Centro de Investigación en Tecnologías de Información y Automatización (CENTIA),
Universidad de las Américas – Puebla (UDLAP),
Ex-Hacienda Santa Catarina Mártir,
Cholula, Puebla, México 72820
{ricardo.perezaa, mariap.gomez, antonio.aguilera}@udlap.mx

**Abstract.** We describe the application of 1-Dimensional Kohonen Networks in the classification of color 2D images which has been evaluated in Popocatépetl Volcano's images. The Popocatépetl, located in the limits of the State of Puebla in México, is active and under monitoring since 1997. We will consider one of the problems related with the question if our application of the Kohonen Network classifies according to the total intensity color of an image or well, if it classifies according to the connectivity, i.e. the topology, between the pixels that compose an image. In order to give arguments that support our hypothesis that our procedures share the classification according to the topology of the pixels in the images, we will present two approaches based a) in the evaluation of the classification given by the network when the pixels in the images are permuted; and,b) when an additional metric to the Euclidean distance is introduced.

## 1 Introduction

It is well known the application of 1-Dimensional Kohonen Networks in the non-supervised classification of data with an elevated redundancy degree [5]. On the other hand, non-supervised image classification is an important vision task where images with similar features are grouped in classes. Many processing tasks (description, object recognition or indexing, for example) are based on such a preprocessing [8]. In this paper, we take in account these ideas in order to apply the methods associated to Kohonen Networks to provide solutions to automatic classification of images. The remainder of this paper is organized as follows: Section 1 describes the basis of the 1-Dimensional Kohonen Networks, Section 2 describes some procedures to take in account in order to avoid training bias, Section 3 describes the procedures and applications related to the classification of 2D color images through Kohonen networks and our results and discussion, finally Section 4 presents conclusions and future work.

M. Lazo and A. Sanfeliu (Eds.): CIARP 2005, LNCS 3773, pp. 271 – 284, 2005.

## 2 Fundamentals of the 1-Dimensional Kohonen Networks

### 2.1 Classifying Points Embedded in a *n*-Dimensional Space Through a 1-Dimensional Kohonen Network

A Kohonen Network with two layers, where the first one corresponds to $n$ input neurons and the second one corresponds to $m$ output neurons ([4] and [7]) can be used to classify points embedded in a $n$-dimensional space in $m$ categories. The input points will have the form $(x_1, \ldots, x_i, \ldots, x_n)$. The total number of connections of the neurons from the input layer to the neurons in the output layer will be $n \times m$ (See Figure 1). Each neuron $j$, $1 \leq j \leq m$, in the output layer will have associated a $n$-dimensional weights vector which describes a representative of class $C_j$. All these vectors will have the form:

$$\text{Output neuron 1: } W_1 = (w_{1,1}, \ldots, w_{1,n})$$
$$\vdots$$
$$\text{Output neuron } m\text{: } W_j = (w_{j,1}, \ldots, w_{j,n})$$

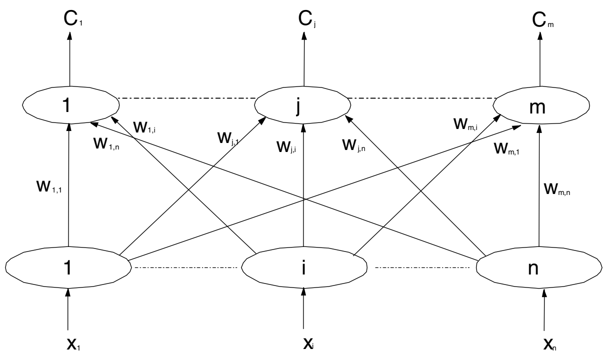



**Fig. 1.** Topology of a 1-dimensional Kohonen Network [5]

### 2.2 Training the 1-Dimensional Kohonen Network

A set of training points are presented to the network $T$ times. According to the literature [5], all the values of the weights vectors can be initialized with random values. In order to determine a winner neuron in the output layer in presentation $t$, $0 \leq t < T$, it is selected that neuron whose weights vector $W_j$, $1 \leq j \leq m$, is the most similar to the input point $P^k$. Such selection is based according to the squared Euclidean distance. The selected neuron will be that with the minimal distance between its weight vector and the input point $P^k$:

$$d_j = \sum_{i=1}^{n} \left(P_i^k - w_{j,i}\right)^2 \qquad 1 \leq j \leq m$$

Once the $j$-th winner neuron in the $t$-th presentation has been identified, its weights are updated according to:

$$w_{j,i}(t+1) = w_{j,i}(t) + \frac{1}{t+1}\left[P_i^k - w_{j,i}(t)\right] \qquad 1 \leq i \leq n$$

When the $T$ presentations have been achieved, the final values of the weights vectors correspond to the coordinates of the 'gravity centers' of the points, or clusters of the $m$ classes.

## 3 Redistribution in the *n*-Dimensional Space of Kohonen Network's Training Set

To avoid training bias, the training data needs to be redistributed. Consider a set of points distributed in a 2D subspace defined by rectangle $[0,1] \times [0,1]$. Moreover, this set of points is embedded in a sub-region delimited, for example, by rectangle $[0.3,0.6] \times [0.3,0.6]$ (Figure 2).

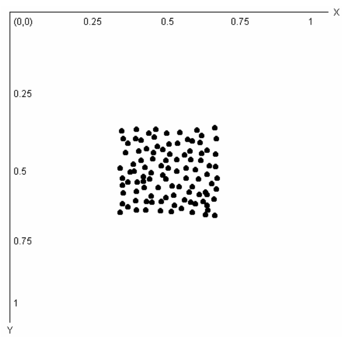


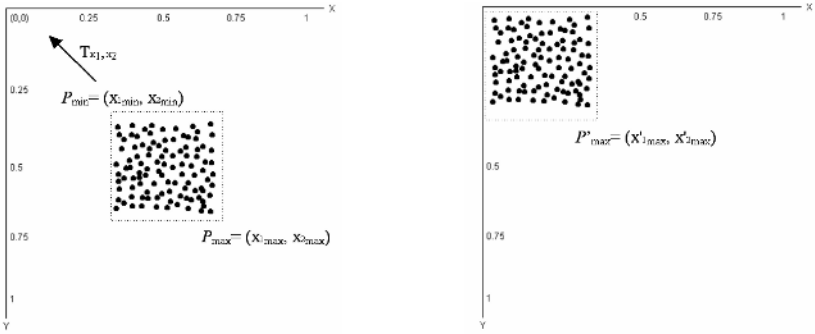

**Fig. 2.** A set of points embedded in $[0.3,0.6] \times [0.3,0.6] \subset [0,1] \times [0,1]$

Because the points are not uniformly distributed in the exemplified 2D space, we can expect important repercussions during their classification process. For example, for a given number of classes, we can obtain some clusters that coincide with other clusters or classes without associated training points. We will describe a simple methodology to distribute uniformly the points of a training set for the general case of a $n$-dimensional space.

Consider a unit $n$-dimensional hypercube $H$ where the points are embedded in their corresponding minimal orthogonal bounding *hyper-box* $h$ such that $h \subseteq H$. The point with the minimal coordinates $P_{\min} = (x_{1_{\min}}, x_{2_{\min}}, \ldots, x_{n-1_{\min}}, x_{n_{\min}})$ and the point with the maximal coordinates $P_{\max} = (x_{1_{\max}}, x_{2_{\max}}, \ldots, x_{n-1_{\max}}, x_{n_{\max}})$ will describe the main diagonal of $h$. We proceed to apply to each point $P = (x_1, x_2, \ldots, x_{n-1}, x_n)$ in the training set, including $P_{\min}$ and $P_{\max}$, the geometric transformation of translation given by:

$$x_1' = x_1 - x_{1_{\min}} \qquad 1 \le i \le n$$

By this way, we will get a new *hyper-box* $h$' and the points that describe the main diagonal of $h$' will be $P'_{\min} = \underbrace{(0,\ldots,0)}_{n}$ and $P'_{\max} = (x'_{1_{\max}}, x'_{2_{\max}}, \ldots, x'_{n-1_{\max}}, x'_{n_{\max}})$. See Figure 3.




a) b)

**Fig. 3.** a) A training set and its minimal orthogonal bounding hyper-box *h*. b) Translation of *h* and the training points such that $P'_{min}$ is the origin of the 2D space.

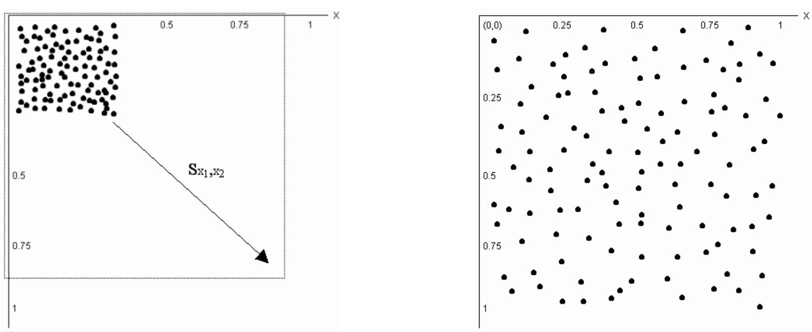
The second part of our procedure will consist in the extension of the current *hyper-box h*' to the whole *n*-dimensional hypercube *H*. The scaling of a point $P = (x_1, x_2, ..., x_{n-1}, x_n)$ is given by multiplying their coordinates by the factors $S_1$, $S_2$, …, $S_n$ each one related with $x_1$, $x_2$, …, $x_n$ respectively in order to produce the new scaled coordinates $x_1'$, $x_2'$, …, $x_n'$ [6]. Because we want to extend the bounding *hyper-box h'* and the translated training points to the whole unit hypercube *H*, we have that by scaling the point $P'_{\max} = (x'_{1_{\max}}, x'_{2_{\max}}, ..., x'_{n-1_{\max}}, x'_{n_{\max}})$ we must obtain the new point $(\underbrace{1,...,1}_{n})$.

That is to say, we define the set of *n* equations:

$$1 = x'_{i_{\max}} \cdot S_i \qquad 1 \le i \le n$$

Starting from these equations we obtain the scaling factors to apply to all points included in the bounding *hyper-box h'* (see Figure 4):

$$S_i = \frac{1}{x'_{i_{\max}}} \qquad 1 \le i \le n$$




a) b)

**Fig. 4.** a) Applying to the translated training set scaling factors such that it will be (b) redistributed to the whole 2D space

Finally, each one of the coordinates in the original points of the training set must be transformed in order to be redistributed in the whole unit $n$-dimensional hypercube $[0,1]^n$ through:

$$x'_i = (x_i - x_{i_{\min}}) \cdot \left( \frac{1}{x'_{i_{\max}}} \right) \qquad 1 \le i \le n$$

# 4 Image Classification Through 1-Dimensional Kohonen Networks

## 4.1 Representing Images Through Vectors in $\mathfrak{R}^n$

Let $m_1$ (rows) and $m_2$ (columns) be the dimensions of a two-dimensional image. Let $n = m_1 \cdot m_2$. Each pixel in the image will have associated a 3-dimensional point ($x_i$, $y_i$, $RGB_i$) such that $RGB_i \in [0, 16777216]$, $1 \le i \le n$, where $RGB_i$ is the color value associated to the $i$-th pixel (assuming that the color of pixels are based in the color model RGB). The color values of the pixels will be normalized such that they will be in [0.0, 1.0] through the transformation:

$$normalized\_RGB_i = \frac{RGB_i}{16777216}$$

Basically, we will define a vector in the $n$-Dimensional space by concatenating the $m_1$ rows in the image considering for each pixel its normalized color RGB value. By this way each image is now associated to a vector in the $n$-dimensional Euclidean space. Because of the color values normalization the scalars in such vectors will be in [0.1]. By this way, a set of training images to be applied in a Kohonen Network will be embedded in an unit $n$-Dimensional hypercube once they have been transformed to their respective associated vectors.

## 4.2 Classifications Results

Our training set contains 148 images selected from *CENAPRED* [3] files. These images represent some of the Popocatépetl volcano fumaroles during the year 2003. The volcano is located in the limits of Puebla state in México; and it is active and under monitoring since 1997. The selected images have an original resolution of 640×480 pixels and 24-bits color under format compression JPG.

We have implemented three 1-Dimensional Kohonen Networks with different topologies (in each case, we applied an scaling to the 148 original images):

- Network Topology $\tau_0$:
  - Images Resolution: 112×64
  - Input Neurons: $n$ = 112×64 = 7,168
  - Output Neurons (classes): $m$ = 20
  - Presentations: $T$ = 10
- Network Topology $\tau_1$:
  - Images Resolution: 56×32
  - Input Neurons: $n$ = 56×32 = 1,792
  - Output Neurons (classes): $m$ = 30
  - Presentations: $T$ = 1,000
- Network Topology $\tau_2$:
  - Images Resolution: 260×180
  - Input Neurons: $n$ = 260×180 = 46800
  - Output Neurons (classes): $m$ = 25
  - Presentations: $T$ = 500

The set of 148 training points (images) were presented the number of times according to the corresponding topologies. The training procedures were applied according to section 2. All the weights vectors' were initialized to 0.5.

Figure 5 shows the classification of the training images using the three proposed topologies. In the figures are also presented the distribution of the 148 training images in each one of the classes. Table 1 presents some images that are representative of each class in Network Topology $\tau_0$ (these images were selected from each class in an arbitrary way).

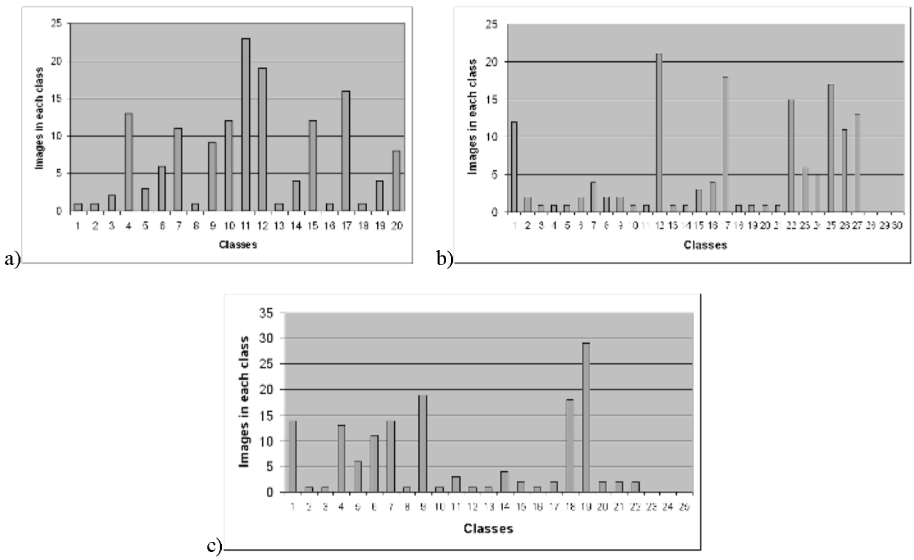



**Fig. 5.** Classification of the 148 training images according to Network Topology a) $\tau_0$, b) $\tau_1$ and c) $\tau_2$

## 4.3 Intensity Based Classification vs. Classification Based in the Topology of Pixels in the Images

One of the problems to consider is related with the question if our implementations of the Kohonen Networks classify according to the total intensity color of an image or well, if they classify according to the connectivity, i.e. the topology, between the pixels that compose an image. In order to give arguments that support our hypothesis that our procedures share the classification according to the topology of the pixels in the images, we have developed two approaches:

- An approach (section 3.3.1) based in a classification of the training images but when their pixels are attached to an specific permutation. If our implementation classify by color intensity, then we can expect a distribution of the images in the classes which would be similar to the distributions presented in Figures 5.

**Table 1.** Representative images of each class in Network Topology $\tau_0$

| | | | | |
|---|---|---|---|---|
|  Class 1 |  Class 2 |  Class 3 |  Class 4 |  Class 5 |
|  Class 6 |  Class 7 |  Class 8 |  Class 9 |  Class 10 |
|  Class 11 |  Class 12 |  Class 13 |  Class 14 |  Class 15 |
|  Class 16 |  Class 17 |  Class 18 |  Class 19 |  Class 20 |

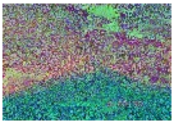
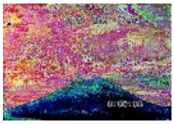
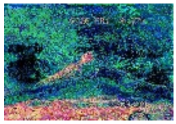
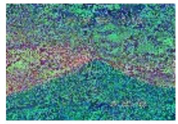
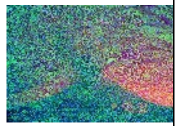
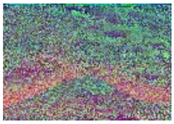
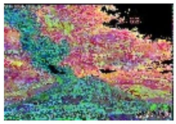
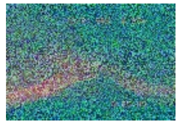
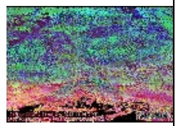
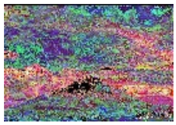
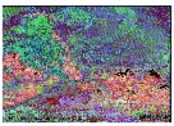
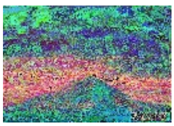
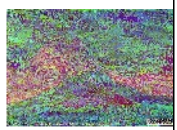
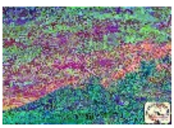
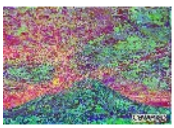
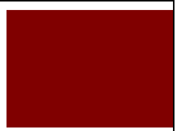
- An approach (section 3.3.2) based in the distances between the weights vectors associated to each output neuron. The clusters themselves are 2D color images if we apply in an inverse way the procedure described in section 3.1. For example, see in the Table 2 the 2D images corresponding to the clusters in Network Topology $\tau_2$. In this approach we will use an additional metric that guarantee the comparison of images only by their color intensity. According to the Kohonen Network training process, the clusters (classes representatives) have been distributed uniformly in an unit *n*-Dimensional hypercube. Such distribution implies, in an implicit way, the fact that each cluster has itself specific characteristics that allow to distinguish its respective class among other classes. By applying the new proposed metric, we can expect that the distances provided by it indicate us a considerable proximity between clusters, hence, they have similar color intensities. Moreover, this last result should establish a considerable distinct distribution respect to the distribution indicated by the Euclidean metric. In the case that our Kohonen Network classify only by color intensity, then the clusters distribution reported by both metrics should be similar.

### 4.3.1 Permutation of Pixels in the Training Images

(See Table 3 for examples of the permutations we describe here.)

- $P_1$: Random permutation of all the pixels in the image.
- $P_2$: Division of the image in 25 rectangular regions and random permutation of the pixels in each region.

**Table 2.** Visualization of clusters in Network Topology $\tau_2$

| | | | | |
|---|---|---|---|---|
|  Class 1 |  Class 2 |  Class 3 |  Class 4 |  Class 5 |
|  Class 6 |  Class 7 |  Class 8 |  Class 9 |  Class 10 |
|  Class 11 |  Class 12 |  Class 13 |  Class 14 |  Class 15 |
|  Class 16 |  Class 17 |  Class 18 |  Class 19 |  Class 20 |
|  Class 21 |  Class 22 |  Class 23 |  Class 24 |  Class 25 |

- $P_3$: Division of the image in 25 rectangular regions and random permutation of such regions.
- $P_4$: Division of the image in 25 rectangular regions and random permutation of the pixels in each region and random permutation of the regions.

Consider to network topology $\tau_1$. In the cases of permutations $P_1$, $P_3$ and $P_4$, we can observe in their corresponding charts (Table 4) the fact that once the training process has finished two classes grouped the 80% of training images. The case of permutation

**Table 3.** Permutations of pixels applied to the training images

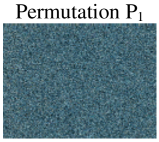
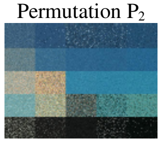
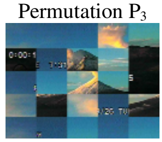
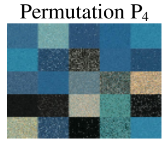
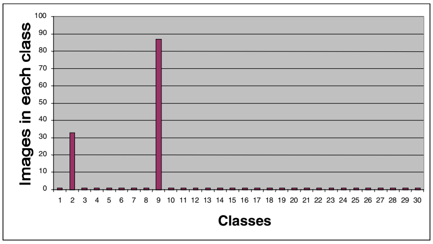
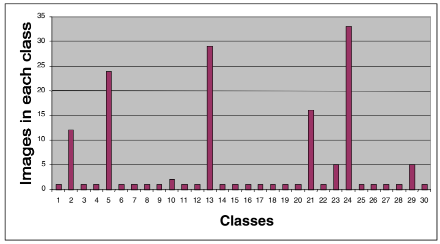
| Original Image | Permutation $P_1$ | Permutation $P_2$ | Permutation $P_3$ | Permutation $P_4$ |
|---|---|---|---|---|
|  |  |  |  |  |

**Table 4.** Distribution of the training images in the classes of network topology $\tau_1$

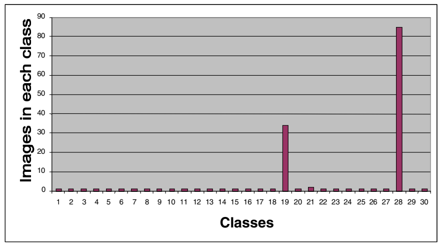



Permutation $P_1$




Permutation $P_2$


Permutation $P_3$




Permutation $P_4$

$\tau_2$ differs from the others by the property that the 80% of training images is grouped in seven classes with more than 5 images each one. From an informal point of view, permutation $P_2$ can be considered visually as a permutation that preserved, compared with the remaining permutations, the connectivity of the pixels respect to the original training images. This is because if we increment the number of rectangular regions (more regions than those in permutation $P_2$) and permute its corresponding pixels, as the number of regions increase the corresponding image will approximate to the original image. In fact, the original images can be seen as images divided in regions with only one pixel each one, obviously, the permutation of the pixel in each region leave to the image in its original state.

### 4.3.2 Analysis Based in an Additional Metric over $\mathfrak{R}^+$

**Definition 1 ([1] & [2]):** Let x, y $\in \mathfrak{R}^+$. Let ρ be the function described as

$$\rho(x,y)=\begin{cases}1-\dfrac{x}{y} & if \quad x<y\\ 1-\dfrac{y}{x} & if \quad y<x\\ 0 & if \quad x=y\end{cases}$$

We will show in Theorem 1 that such function is a metric over $\mathfrak{R}^+$. Appendix A contains the propositions that support our proof.

**Theorem 1:** Let x, y $\in \mathfrak{R}^+$. Therefore ρ(x, y) is a metric over $\mathfrak{R}^+$.
Proof: Let x, y, z $\in \mathfrak{R}^+$.

- We will show that $\rho(x, y) = \rho(y, x)$.
  - If $x = y \Rightarrow \rho(x, y) = 0 = \rho(y, x)$.
  - If $x < y \Rightarrow \rho(x, y) = 1 - x/y = \rho(y, x)$.
  - If $y < x \Rightarrow \rho(x, y) = 1 - y/x = \rho(y, x)$.

  $\therefore (\forall x, y \in \Re^+)(\rho(x, y) = \rho(y, x))$
- By definition of $\rho$: $(\forall x \in \Re^+)(\rho(x, x) = 0)$.
- By definition of $\rho$, if $\rho(x, y) = 0 \Rightarrow x = y$.
- By property A.1, $(\forall x, y \in \Re^+)(\rho(x, y) \geq 0)$.
- We will show that $\rho(x, z) \leq \rho(x, y) + \rho(y, z)$.
  - If $x = y = z \Rightarrow \rho(x, z) = 0 \leq \rho(x, y) + \rho(y, z) = 0$
  - If $x = z, x \neq y \Rightarrow \rho(x, z) = 0 \leq \rho(x, y) + \rho(y, z)$
  - If $x = y, x \neq z \Rightarrow \rho(x, z) = \rho(y, z)$
  - If $y = z, x \neq y \Rightarrow \rho(x, z) = \rho(x, y)$
  - If $x < y < z \Rightarrow$ By lemma A.1, $\rho(x, z) < \rho(x, y) + \rho(y, z)$
  - If $x < z < y \Rightarrow$ By lemma A.2, $\rho(x, z) < \rho(x, y) + \rho(y, z)$
  - If $z < x < y \Rightarrow$ By lemma A.3, $\rho(x, z) < \rho(x, y) + \rho(y, z)$
  - If $z < y < x \Rightarrow$ By lemma A.4, $\rho(x, z) < \rho(x, y) + \rho(y, z)$
  - If $y < x < z \Rightarrow$ By lemma A.5, $\rho(x, z) < \rho(x, y) + \rho(y, z)$
  - If $y < z < x \Rightarrow$ By lemma A.6, $\rho(x, z) < \rho(x, y) + \rho(y, z)$

  $\therefore (\forall x, y, z \in \Re^+)(\rho(x, z) \leq \rho(x, y) + \rho(y, z))$

$\therefore$ $\rho$ is a metric over $\Re^+$. ◽

Let I be an image. We know that each one of its pixels $p_i$ will have associated a vector $(x_i, y_i, RGB_i)$, $i \in [1, n]$, $RGB_i \in [0, 16777216]$. Lets assume that the dimensions of each pixel are equal to one. We will define to the Total Intensity of I, denoted by T(I), as follows:

$$T(I) = \sum_{i=1}^{n} RGB_i$$

Let $I_A$ and $I_B$ two images with the same geometrical dimensions. Let $T(I_A)$ and $T(I_B)$ their corresponding Total Intensities. Because $T(I_A), T(I_B) \in \Re^+$ we can determine its distance through the metric $\rho$.

Now, we will define the similarity between images $I_A$ and $I_B$ according to the value of $\rho(T(I_A), T(I_B))$. Let $0 \leq \varepsilon < 1$ be an arbitrary value such that we will establish

$$I_A \text{ is similar to } I_B \Leftrightarrow \rho(T(I_A), T(I_B)) < \varepsilon$$

A classification based in metric $\rho$ will not take in account the connectivity between the pixels in the images. For example, for the images presented in Figure 6 we have that $\rho(T(I_A), T(I_B)) = 0$.

The Kohonen Network we implemented uses as part of its processes of training and classification the Euclidean metric over $\Re^n$. Because each one of the representatives of the classes (clusters) in the network are themselves vectors in $\Re^n$, then we can determine the Euclidean distance between any pair of clusters.


$I_A$


$I_B$

**Fig. 6.** An example where $\rho(T(I_A), T(I_B)) = 0$.$I_B$ is image $I_A$ with permutation $P_3$.

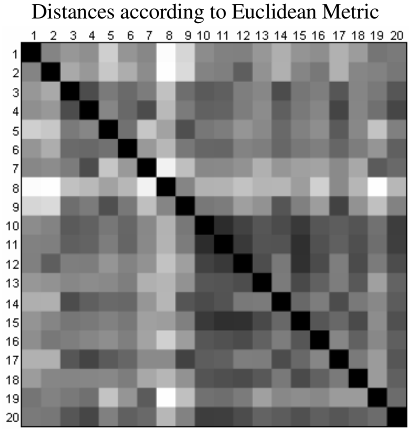
We will define a false color map that will represent the distribution of the clusters in the subspace $[0, 1]^n$. The maximal Euclidean distance between any two clusters will be $d_{\max} = \sqrt{n}$ while in the other hand the minimal distance will be $d_{min} = 0$. Every Euclidean distance between two clusters will be associated with a color in the grayscale through $\frac{d}{d_{\max}} \cdot 256$. By this way if $d = 0$ then it will have associated the black color while if $d = d_{max}$ then it will have associated the white color.

Moreover, we will define a false color map that represent the distances between the clusters in the subspace $[0, 1]^n$ under our metric $\rho$. For any clusters $a$ and $b$, $\rho(a, b)$ will be associated with the grayscale through $\rho(a, b)\cdot 256$. If $\rho(a, b) = 0$ then $a = b$ and therefore such distance will be represented through the black color. On the other hand, $\rho(a, b)\cdot 256 \rightarrow 256$ while $\rho(a, b) \rightarrow 1$.

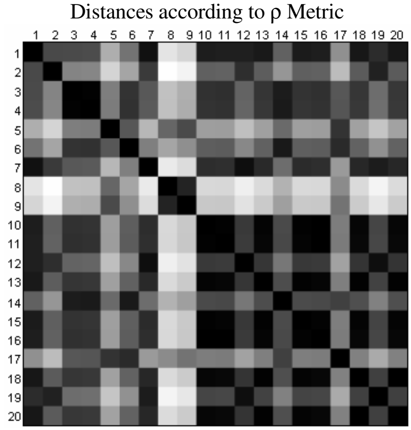
Consider Network Topology $\tau_0$. The false color maps associated to the distances between the clusters under the Euclidean metric and $\rho$ metric are presented in Table 5. It can be observed in the map under metric $\rho$ that the 47% of the distances between clusters are less than 0.20. This indicates that according this metric an important

**Table 5.** False Color Maps that show the distances between clusters in Network Topology $\tau_0$

Distances according to Euclidean Metric




Distances according to ρ Metric




In both maps:

- (white box) Maximum Distance
- (black box) Minimum Distance

number of clusters are similar with $\varepsilon = 0.20$ (in fact the mean distance in this metric was 0.2542 with variance 0.0373 and standard deviation 0.1933). In the other hand, we have that for topology $\tau_0$ $n = 7168$, hence, $d_{max} = \sqrt{7168} = 84.66$. Analogously we consider the number of distances whose value is less than the 20% of $d_{max}$. By this way, the map based in the Euclidean metric reports that only the 19% of the distances between clusters are lower than 16.9328 (the mean distance under Euclidean metric was 24.2119 with variance 94.7531 and standard deviation 9.7341). In conclusion, both metrics report different distributions of the clusters which makes visible the differences between a classification based in topology of pixels, by the Kohonen Network, and a classification based in color intensities of the images.

## 5 Conclusions

According to the results provided by the approaches discussed in sections 3.3.1 and 3.3.2 we can infer that image classification based in a 1-Dimensional Kohonen Network groups an images set according to features based in the connectivity between pixels, i.e., their topology. As part of our future work, we will analyze in a detailed way the images contained in each one of our classes and their respective neighborhoods in order to determine some features shared by these images. By identifying these features, in our images domain, we will analyze the possible application of our classifications in the prediction of events of Popocatépetl volcano.

Another objective, with respect to future work, considers the comparison of our presented procedures, based in a non-supervised classification, with other techniques that allow the automated retrieval and classification of images such as Case Based Reasoning (CBR) and Image Based Reasoning (IBR).

## References

1. Aguilera, Antonio; Lázzeri Menéndez, Santos Gerardo & Pérez-Aguila, Ricardo. Image Based Reasoning Applied to the Comparison of the Popocatépetl Volcano's Fumaroles. Proceedings of the IX Ibero-American Workshops on Artificial Intelligence, Iberamia 2004, pp. 3-8. ISBN: 968-863-783-6. November 22 to 23, 2004. National Institute of Astrophysics, Optics and Electronics (INAOE), Puebla, México.
2. Aguilera, Antonio; Lázzeri Menéndez, Santos Gerardo & Pérez-Aguila, Ricardo. A Procedure for Comparing Color 2-Dimensional Images Through their Extrusions to the 5-Dimensional Colorspace. Proceedings of the 15th International Conference on Electronics, Communications, and Computers CONIELECOMP 2005, pp. 300-305. Published by the IEEE Computer Society. ISBN: 0-7695-2283-1. February 28 to March 2, 2005. Puebla, México.
3. CENAPRED (Centro Nacional de Prevención de Desastres), México. Web Site: http://www.cenapred.unam.mx (April 2003).
4. Davalo, Eric & Naïm, Patrick. Neural Networks. The Macmillan Press Ltd, 1992.
5. Hilera, José & Martínez, Victor. Redes Neuronales Artificiales. Alfaomega, 2000. México.

6. Pérez Aguila, Ricardo. The Extreme Vertices Model in the 4D space and its Applications in the Visualization and Analysis of Multidimensional Data Under the Context of a Geographical Information System. Thesis for the Master's Degree in Sciences. Universidad de las Américas - Puebla. Puebla, México, 2003.
7. Ritter, Helge; Martinetz, Thomas & Schulten, Klaus. Neural Computation and Self-Organizing Maps, An introduction. Addison-Wesley Publishing Company, 1992.
8. Zerubia, Josiane; Yu, Shan; Kato, Zoltan & Berthod, Mark. Bayesian Image Classification Using Markov Random Fields. Image and Vision Computing, 14:285-295, 1996.

## Appendix A: Properties of ρ Function

**Property A.1:** Let $x, y \in \Re^+$. Therefore $\rho(x, y) \in [0, 1)$.

Proof: We consider three cases,

- If $x = y \Rightarrow \rho(x, y) = 0$.
- If $x < y \Rightarrow 1 > x/y > 0 \Rightarrow -1 < -x/y < 0 \Rightarrow 0 < 1 - x/y < 1 \Rightarrow 0 < \rho(x, y) < 1$.
- If $x > y \Rightarrow 1 > y/x > 0 \Rightarrow -1 < -y/x < 0 \Rightarrow 0 < 1 - y/x < 1 \Rightarrow 0 < \rho(x, y) < 1$.

$\therefore (\forall x, y \in \Re^+)(\rho(x, y) \in [0,1))$. ◘

**Lemma A.1:** Let $x, y, z \in \Re^+$ such that $x < y < z$. Then $\rho(x, z) < \rho(x, y) + \rho(y, z)$.

Proof: By definition 1 and considering the established hypothesis we have that

$$\rho(x, y) = 1 - x/y \qquad \rho(y, z) = 1 - y/z \qquad \rho(x, z) = 1 - x/z$$

Because by hypothesis, $x < y \Rightarrow x/z < y/z \Rightarrow -x/z > -y/z \Rightarrow 1 - x/z > 1 - y/z$
$\Rightarrow \rho(x, z) > \rho(y, z)$.

Because by hypothesis, $y < z \Rightarrow x/y > x/z \Rightarrow -x/y < -x/z \Rightarrow 1 - x/y < 1 - x/z$
$\Rightarrow \rho(x, y) < \rho(x, z)$.

Due to $1 > \rho(x, z) > \rho(y, z)$ and $1 > \rho(x, z) > \rho(x, y)$
$\Rightarrow 2 > 2\rho(x, z) > \rho(x, y) + \rho(y, z) = 2 - (x/y + y/z) \Rightarrow 2 > 2 - (x/y + y/z)$
$\Rightarrow 1 > 1 - (x/y + y/z) \Rightarrow 2 > \rho(x, y) + \rho(y, z) > 1 > 1 - (x/y + y/z)$
$\Rightarrow \rho(x, y) + \rho(y, z) > 1 \Rightarrow \rho(x, y) + \rho(y, z) > 1 > \rho(x, z)$.
$\therefore \rho(x, z) < \rho(x, y) + \rho(y, z)$. ◘

**Lemma A.2:** Let $x, y, z \in \Re^+$ such that $x < z < y$. Then $\rho(x, z) < \rho(x, y) + \rho(y, z)$.

Proof: By definition 1 and considering the established hypothesis we have that

$$\rho(x, y) = 1 - x/y \qquad \rho(y, z) = 1 - z/y \qquad \rho(x, z) = 1 - x/z$$

Because by hypothesis, $x < z \Rightarrow x/y < z/y \Rightarrow -x/y > -z/y \Rightarrow 1 - x/y > 1 - z/y$
$\Rightarrow \rho(x, y) > \rho(y, z)$.

Because by hypothesis, $z < y \Rightarrow x/z > x/y \Rightarrow -x/z < -x/y \Rightarrow 1 - x/z < 1 - x/y$
$\Rightarrow \rho(x, z) < \rho(x, y)$.

Due to $\rho(x, z) < \rho(x, y)$ and $\rho(y, z) < \rho(x, y)$ $\therefore \rho(x, z) < \rho(x, y) + \rho(y, z)$. ◘

**Lemma A.3:** Let $x, y, z \in \Re^+$ such that $z < x < y$. Then $\rho(x, z) < \rho(x, y) + \rho(y, z)$.

Proof: By definition 1 and considering the established hypothesis we have that

$$\rho(x, y) = 1 - x/y \qquad \rho(y, z) = 1 - z/y \qquad \rho(x, z) = 1 - z/x$$

Because by hypothesis, $x < y \Rightarrow z/x > z/y \Rightarrow -z/x < -z/y \Rightarrow 1 - z/x < 1 - z/y$
$\Rightarrow \rho(x, z) < \rho(y, z) \Rightarrow \rho(x, z) < \rho(y, z) < \rho(x, y) + \rho(y, z)$
$\therefore \rho(x, z) < \rho(x, y) + \rho(y, z)$. ◘

**Lemma A.4:** Let x, y, z ∈ $\Re^+$ such that z < y < x. Then ρ(x, z) < ρ(x, y) + ρ(y, z).

Proof: By definition 1 and considering the established hypothesis we have that

$$\rho(x, y) = 1 - y/x \qquad \rho(y, z) = 1 - z/y \qquad \rho(x, z) = 1 - z/x$$

Because by hypothesis, $z < y \Rightarrow z/x < y/x \Rightarrow -z/x > -y/x \Rightarrow 1 - z/x > 1 - y/x$
$\Rightarrow \rho(x, z) > \rho(x, y)$.

Because by hypothesis, $y < x \Rightarrow z/y > z/x \Rightarrow -z/y < -z/x \Rightarrow 1 - z/y < 1 - z/x$
$\Rightarrow \rho(y, z) < \rho(x, z)$.

Due to $1 > \rho(x, z) > \rho(y, z)$ and $1 > \rho(x, z) > \rho(x, y)$
$\Rightarrow 2 > 2\rho(x, z) > \rho(x, y) + \rho(y, z) = 2 - (y/x + z/y) \Rightarrow 2 > 2 - (y/x + z/y)$
$\Rightarrow 1 > 1 - (y/x + z/y) \Rightarrow 2 > \rho(x, y) + \rho(y, z) > 1 > 1 - (y/x + z/y)$
$\Rightarrow \rho(x, y) + \rho(y, z) > 1 \Rightarrow \rho(x, y) + \rho(y, z) > 1 > \rho(x, z)$
$\therefore \rho(x, z) < \rho(x, y) + \rho(y, z)$. ◘

**Lemma A.5:** Let x, y, z ∈ $\Re^+$ such that y < x < z. Then ρ(x, z) < ρ(x, y) + ρ(y, z).

Proof: By definition 1 and considering the established hypothesis we have that

$$\rho(x, y) = 1 - y/x \qquad \rho(y, z) = 1 - y/z \qquad \rho(x, z) = 1 - x/z$$

Because by hypothesis, $y < x \Rightarrow y/z < x/z \Rightarrow -y/z > -x/z \Rightarrow 1 - y/z > 1 - x/z$
$\Rightarrow \rho(y, z) > \rho(x, z) \Rightarrow \rho(x, z) < \rho(y, z) < \rho(x, y) + \rho(y, z)$
$\therefore \rho(x, z) < \rho(x, y) + \rho(y, z)$. ◘

**Lemma A.6:** Let x, y, z ∈ $\Re^+$ such that y < z < x. Then ρ(x, z) < ρ(x, y) + ρ(y, z).

Proof: By definition 1 and considering the established hypothesis we have that

$$\rho(x, y) = 1 - y/x \qquad \rho(y, z) = 1 - y/z \qquad \rho(x, z) = 1 - z/x$$

Because by hypothesis, $y < z \Rightarrow y/x < z/x \Rightarrow -y/x > -z/x \Rightarrow 1 - y/x > 1 - z/x$
$\Rightarrow \rho(x, y) > \rho(x, z) \Rightarrow \rho(x, z) < \rho(x, y) < \rho(x, y) + \rho(y, z)$
$\therefore \rho(x, z) < \rho(x, y)$. ◘