Experimental results of an Evolution-based Adaptation Strategy for VQ Image Filtering.

A. I. González[1], M. Graña[1], J. Ruiz Cabello[2], A. D'Anjou[1], F.X. Albizuri[1]

[1]Dpto. C.C.I.A. Univ. del País Vasco / EHU
Apdo. 649, 20080 - San Sebastián, Spain
ccpgrrom@si.ehu.es

[2]Unidad de RMN, Universidad Complutense
Paseo Juan XXIII, 1, Madrid 28040, Spain
ruizcabe@eucmax.sim.ucm.es

ABSTRACT. *In this paper we propose the filtering of images based on the codebooks obtained from an Evolution-based Adaptation Strategy for Vector Quantization (VQ). This evolution based VQ Bayesian Filter (VQBF) is applied to noise removal and segmentation of a high-resolution Magnetic Resonance Image. We compare our approach with other more conventional smoothing filters. The results show that VQBF performs a smoothing that preserves region boundaries and small details. It does not show the strong boundary diffusion and displacement that are common to smoothing filters. Border detection on the filtered images is also presented.*

Keywords: Evolution Strategies, Clustering, Noise Removal, Image Segmentation, Bayesian Image Processing.

1.- INTRODUCTION.

Vector Quantization (VQ) [1, 2, 3] is the process of replacing the signal vectors by representatives chosen from a set named codebook. Its main application is in signal compression, although sometimes it is used as a feature extraction method for classification or as a signal filtering method [4]. The work proposed in this paper falls into this last family of VQ applications. We follow an uncommon approach in the way we decompose and process an image, which we name VQ Bayesian filter (VQBF). Given a codebook, VQBF consists in two steps: (1) we determine the codevector that encodes the vector given by a pixel and its neighborhood, and (2) we substitute the pixel by the central element in the codevector. The encoding process is a Maximum A Posteriori (MAP) classification and, thus, VQBF performs a kind of Bayesian image processing [5, 6]. In this context (1) the stochastic model and its estimation are, respectively, the codebook and the search for the optimal codebook; and (2) the process does not involve complex and lengthy relaxation procedures (i.e.: simulated annealing [5]), only the search for the nearest codevector.

Our approach share the general problem of codebook design, which is a kind of clustering problem. Between the possible techniques that can be used to compute the codebook there are

some methods based on evolutionary algorithms [7-20]. The common approach of all these works is the mapping of complete clustering solutions to population individuals. The fitness function is the chosen ad-hoc clustering criterion function. The authors propose a wide variety of representations of clustering solutions as population individuals, ranging from the set of cluster representatives to the membership (hard or fuzzy) matrices of the clusters. Evolution operators, recombination and mutation, are defined suitably to be closed operators on the chosen representation.

In this respect, our own contribution [21] consist in the proposition of an Evolution-based Adaptation Strategy which shows the following features: (1) vector real valued individuals, (2) the main genetic operator is mutation, and (3) mutation is based in individuals local information. We assume in our algorithm that the population is the codebook given by a set of codevectors, which induce a Voronoi partition over the input space, and a clustering of the sample. The mutation operator is guided by the estimated covariance matrices of the clusters. We have not defined any cross or recombination operators. The selection operator extracts the next generation population from the pool of parents and offspring generated by mutation. We have worked with two types of selection operators one that selects a fixed preset number of individuals for the next population, and other one that tries to determine the optimal number of individuals.

In Section 2, we present the VQ Evolution-based Adaptation Strategy. Section 3 presents the VQBF more formally. Section 4 presents the results on the application to an image and comparison with other conventional approaches. Finally, Section 5 gives some concluding remarks.

2.- THE EVOLUTION-BASED ADAPTATION STRATEGY.

In this section we review the Evolution-based Adaptation Strategy proposed in [21] as will be used in this work. The main addition to the basic definition is the attempt to define the selection of an optimally sized population. A widely accepted [22] pseudocode representation of the general structure of the algorithm of Evolution Strategies is given in figure 1. The Evolution-based Adaptation Strategy proposed is heavily influenced by the use of the Euclidean distance, the consideration of other clustering measures will imply that some algorithm elements must be redefined. We will start describing in detail the elements of the Evolution-based Adaptation Strategy.

```
t:= 0
initialize P(t)
evaluate P(t)
while not terminate do
    P'(t):= recombine P(t)
    P''(t):= mutate P'(t)
    evaluate P''(t)
    P(t+1):= select (P''(t) U Q)
    t:= t+1
end while
```

Figure 1. General structure of an Evolution-based Adaptation Strategy.

2.1. Individuals, Population and Fitness functions.

We make each individual to correspond to a single cluster center. A single solution to the VQ problem is mapped into the entire population. The population at generation **t** is given by

$$P(\mathbf{t}) = \left\{ \mathbf{y}_i(\mathbf{t}) \mid \mathbf{y}_i(\mathbf{t}) \in \Re^d ; i = 1...c \right\} \tag{1}$$

The population size *c* corresponds to the number of clusters searched in the data. We have not included mutation parameters in the definition of the individuals, because we will use for this role the covariance matrices computed over the sample data.

The local fitness of each individual is its local quantization error relative to the sample data.

$$F_i(\boldsymbol{t}) = \sum_{j=1}^{n} \left\| \mathbf{x}_j - \mathbf{y}_i(\boldsymbol{t}) \right\|^2 \delta_{ij}(\boldsymbol{t}) \tag{2}$$

We used to compute this fitness a sample data $\aleph = \left\{ \mathbf{x}_1, ..., \mathbf{x}_n \mid \mathbf{x}_i \in \Re^d \right\}$ extracted randomly from the original data set. As the individuals do not specify complete clustering solutions, we must consider a fitness function for the population as a whole. This population fitness corresponds to the objective function to be minimized, because it is the whole population, which specifies the clustering solution and can be evaluated as

$$F(\boldsymbol{t}) = \sum_{i=1}^{c} F_i(\boldsymbol{t}) \tag{3}$$

Our population fitness corresponds to the within cluster scatter of the clustering specified by the population $S_W = F(\boldsymbol{t})$. The well known equation relating the within and between cluster scattering [23]: $S = S_W + S_B$, can be written using (2) and (3) as:

$$S = \sum_{j=1}^{n} \left\| \mathbf{x}_j - \bar{\mathbf{y}} \right\|^2 = \sum_{i=1}^{c} F_i(\boldsymbol{t}) + \sum_{i=1}^{c} \left\| \mathbf{y}_i(\boldsymbol{t}) - \bar{\mathbf{y}} \right\|^2 \tag{4}$$

Where $\boldsymbol{S}$ remains constant for the same data sample, and $\bar{\mathbf{y}}$ denotes the centroid of the entire data sample $\aleph$. The constraint specified in (4) implies that the isolated minimization of individual fitness would lead to the population fitness minimization.

2.2. The mutation operator.

The recombination operators found in the literature of Evolution Strategies do not look as appropriate sources for new cluster representatives, therefore we have not defined any recombination operator. Only the mutation operator introduces evolutive changes. As is customary in Evolution Strategies, our mutation operator is a random perturbation that follows a zero-mean normal distribution.

The set of mutation parents is composed of the individuals whose local fitness is greater than the mean of the local fitness in its generation. Formally, this set is given by:

$$\phi(\boldsymbol{t}) = \left\{ i \middle| F_i(\boldsymbol{t}) \geq \bar{F}(\boldsymbol{t}) \right\} \text{ where } \bar{F}(\boldsymbol{t}) = \frac{1}{c} \sum_{i=1}^{c} F_i(\boldsymbol{t}) \tag{5}$$

We will be allowed a number of mutations that we have decided to approach as much as possible to a fixed number of mutations m, so that the number of mutations per individual $m_i(\mathbf{t})$ will depend on the size of $\phi(\mathbf{t})$,

$$m_i(\boldsymbol{t}) = \left\lceil \frac{m}{|\phi(\boldsymbol{t})|} \right\rceil \tag{6}$$

The information used to compute mutations is the local covariance matrices of the sample partition associated with each individual, so that the mutation operator is naturally adapted to each individual. The expression of the local covariance matrices is

$$\hat{\Sigma}_i(\boldsymbol{t}) = (n-1)^{-1} \sum_{j=1}^{n} \left(\mathbf{x}_j - \mathbf{y}_i(\boldsymbol{t}) \right) \left(\mathbf{x}_j - \mathbf{y}_i(\boldsymbol{t}) \right)' \delta_{ij}(\boldsymbol{t}) \tag{7}$$

Mutations are computed along the axes defined by the eigenvectors of the estimated local cluster covariance matrix. The number of mutations $m_{ij}(\boldsymbol{t})$ along each eigenaxis is proportional to the relative magnitude of its eigenvalue. The width of mutation perturbations α_k are subject to an annealing process $C(\boldsymbol{t})$ that reduces its scale in each generation. We assume that there is a monotonic improvement in the solution. Let $\Lambda_i = \text{diag}\left(\lambda_{ij}, j = 1...d\right)$ and $\Phi_i = \left[\mathbf{e}_{ij}, j = 1...d\right]$ denote, respectively, the eigenvalue and eigenvector matrices of $\hat{\Sigma}_i(\boldsymbol{t})$. Then the set of mutations generated along the axis defined by eigenvector $\mathbf{e}_{ij}$ is:

$$P''_{ij}(\boldsymbol{t}) = \left\{ \mathbf{y}_i \pm \alpha_k \lambda_{ij} \mathbf{e}_{ij} \middle| k = 1..m_{ij}(\boldsymbol{t}), i \in \phi(\boldsymbol{t}) \right\}$$
$$m_{ij}(\mathbf{t}) = \text{round}\left(\frac{m_i(\mathbf{t})\lambda_{ij}}{2\sum_{l=1}^{d} \lambda_{il}} \right), \alpha_k = \frac{C(\boldsymbol{t})k}{m_{ij}(\boldsymbol{t})} \tag{8}$$

The set of individuals generated by the mutation operator is

$$P''(\boldsymbol{t}) = \bigcup_{i,j} P''_{ij}(\boldsymbol{t}) \tag{9}$$

2.3. The selection operator.

The remaining operator to be defined is the selection operator, which determines the individuals of the population for the next generation. In the definition of this operator we have followed the so called (μ+λ)-strategy [24]. We pool together parents and children, so that $Q = P(\boldsymbol{t})$. In our case, μ is the number of classes, and λ the number of generated mutations.

We have defined two selector operators, one that selects a fixed number *c* of the best individuals for the next generation based on individual distortion functions. And a second one that also determines the number of individuals in the next generation. The later one employs a fitness function based on individual distortion and entropy.

2.3.1 Selection of a fixed size population.

Selection can not be based on the original individual fitness functions $F_i(\boldsymbol{t})$ because they do not carry information about the interaction effects introduced by new individuals generated by mutation. The optimal approach to the implementation of this selection operator consists in computing the fitness of all the possible populations of size *c* extracted from $P''(\boldsymbol{t}) \cup P(\boldsymbol{t})$. The complexity of this approach is of the order of

$$\binom{|P''(\mathbf{t}) \cup P(\mathbf{t})|}{\mathrm{c}}$$

computations of population fitness functions. This computational burden largely questions the feasibility of this approach in any practical application. Therefore, we have work with an alternative selection operator to reduce the complexity combinatorial growth.

We pool together the parents and the individuals generated by mutation. Let us denote as $F^S(\boldsymbol{t})$ the fitness of the population $P''(\boldsymbol{t}) \cup P(\boldsymbol{t})$. A way to measure the importance of a given individual is to compute the effect of removing it from the population. That is, we compute $F_i^S(\boldsymbol{t})$ as the fitness of the population $P''(\mathbf{t}) \cup P(\mathbf{t}) - \{\mathbf{y}_i(\mathbf{t})\}$ for each $\mathbf{y}_i(\mathbf{t}) \in P''(\mathbf{t}) \cup P(\mathbf{t})$. The

importance of the individual would be measured by $F_i^s(t) - F^s(t)$. As $F_i^s(t) \geq F^s(t)$ for all the individuals, it suffices to compute $F_i^s(t)$ to measure the *importance* of the individual. Notice that for empty cluster representatives, which can be the case of some mutation generated individuals, their significance is null $F_i^s(t) = F^s(t)$, so that they will be discarded automatically. It is trivial to verify that no empty cluster will be selected using this fitness function, unless there is someone in the original population and all the mutations generate empty cluster representatives.

For notation simplicity, let $\lambda = |P''(t)|$ be the number of individuals effectively generated by mutation. A formal definition of the individual fitness function used by the selection operator is as follows:

$$F_i^s(t) = \sum_{\substack{k=1 \\ i \neq k}}^{c+\lambda} \sum_{j=1}^{n} \left\| \mathbf{x}_j - \mathbf{y}_k(t) \right\|^2 \delta_{kj}^s(t)$$

$$\delta_{kj}^s(t) = \begin{cases} 1 & k = \underset{\substack{l=1,..c+\lambda \\ l \neq i}}{\operatorname{argmin}} \left\{ \left\| \mathbf{x}_j - \mathbf{y}_l(t) \right\|^2 \right\} \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

The selection operator selects the c best individual's fitness to obtain the next generation population. Formally:

$$P(t+1) = \mathbf{select}\left(P''(t) \cup Q\right) \;=\; \left\{ \mathbf{y}_i \in P^*(t);\ i = 1,...,c \right\}$$

$$P^*(t) = \left\{ \mathbf{y}_{i_1}, ..., \mathbf{y}_{i_{c+\lambda}} \mid i_j < i_k \Rightarrow F_j^s(t) > F_k^s(t) \right\} \tag{11}$$

We introduce an elitist criterion, to ensure convergence, as follows: the selected population is accepted if it is better than the previous one.

The computation requirements of this selection operator are linear in the number of cluster representatives, and it can be easily speed up using the simple programming trick of precomputing the two nearest cluster representatives. Although, this selection operator is clearly suboptimal, the experimental works assess the positive balance between its suboptimality versus its computational efficiency.

2.3.2 Selection of an optimally sized population.

Like in the selection operator defined above, this variation of the selection can not be based on the original individual fitness functions $F_i(t)$; we need more information about the interaction

effects introduced by the mutation generated individuals and how to determine the number of individuals that will form the next population. The optimal approach to the implementation of this selection operator would consist in computing the fitness of all the possible populations of varying size k (k = 1,… , μ+λ) extracted from $P''(\boldsymbol{t}) \cup P(\boldsymbol{t})$. The complexity of this process would be of the order of

$$\sum_{k=1}^{\mu+\lambda} \binom{|P'(\mathbf{t}) \cup P(\mathbf{t})|}{\mathrm{k}}$$

computations of population fitness functions. Again, we work with an alternative selection operator of reduced complexity.

We pool together the parents and the individuals generated by mutation. Let us denote as $F^s(\boldsymbol{t})$ the fitness of the population $P''(\boldsymbol{t}) \cup P(\boldsymbol{t})$. Now, we define $F_i^s(\boldsymbol{t})$ as a linear combination of local distortion and entropy increment for each individual i. We previously normalize into the interval [*0*, *1*] both terms. Depending on the relative weight η, the selection operator gives more priority to the local distortion or to the entropy increment:

$$F_i^s(\boldsymbol{t}) = D_i^2(\boldsymbol{t}) - \eta \cdot \Delta H_i(\boldsymbol{t}) \tag{12}$$

The computation of the local distortion $D_i^2(\boldsymbol{t})$ is identical to the individual fitness of the previous section, let $\lambda = |P''(\boldsymbol{t})|$:

$$D_i^2(\boldsymbol{t}) = \sum_{\substack{k=1 \\ i \neq k}}^{\mu+\lambda} \sum_{j=1}^{n} \left\| \mathbf{x}_j - \mathbf{y}_k(\boldsymbol{t}) \right\|^2 \delta_{kj}^s(\boldsymbol{t}) \tag{13}$$

$$\delta_{kj}^s(\boldsymbol{t}) = \begin{cases} 1 & k = \underset{\substack{l=1,..\mu+\lambda \\ l \neq i}}{\operatorname{argmin}} \left\{ \left\| \mathbf{x}_j - \mathbf{y}_l(\boldsymbol{t}) \right\|^2 \right\} \\ 0 & \text{otherwise} \end{cases}$$

The entropy of each individual $\mathrm{H_i}(\boldsymbol{t})$ is computed with the same strategy followed to compute the local distortion: we compute the entropy of the population $P''(\mathbf{t}) \cup P(\mathbf{t}) - \{\mathbf{y}_i(\mathbf{t})\}$ for each $\mathbf{y}_i(\mathbf{t}) \in P''(\mathbf{t}) \cup P(\mathbf{t})$. The individual entropy increment would be measured by $\Delta H_i(\boldsymbol{t}) = H(\boldsymbol{t}) - H_i(\boldsymbol{t})$, where $H(\boldsymbol{t})$ is the entropy of the whole population. Let n_i the number of data items classified into individual i, then we calculate $H_i(\boldsymbol{t})$ as:

$$H_i(\boldsymbol{t}) = \sum_{\substack{k=1 \\ i \neq k}}^{\mu+\lambda} \frac{1}{n_i} \log \frac{1}{n_i} \tag{14}$$

The selection operator iteratively adds the best individual until a stopping condition is met. This stopping condition is defined over the relative increment of the accumulated fitness functions of the selected individuals that can be above a predetermined threshold T:

$$P(t+1) = \mathbf{select}\left(P''(t) \cup Q\right) = \left\{ \mathbf{y}_{i_1}(t),\ldots,\mathbf{y}_{i_z}(t) \mid i_j < i_{j+1} \Rightarrow F_j^s(t) > F_{j+1}^s(t) \wedge \left(\left| \left(\sum_{m=1}^{j+1} F_m^s(t) - \sum_{n=1}^{j} F_n^s(t) \right) \Big/ \sum_{n=1}^{j} F_n^s(t) \right| < T \right) \right. \tag{15}$$

3.- VECTOR QUANTIZATION FOR IMAGE FILTERING.

Besides its known applications in compression VQ has been proposed for digital image processing [4]. It has been suggested that the encoding/decoding introduce some non-linear smoothing of the image that removes some kinds of noise, especially speckle noise. A conventional definition of Vector Quantization [1] is as follows: given a stochastic process $\{\mathbf{X}_t; t>0\}$ whose state space is the d-dimensional (Euclidean) real space R^d, the Vector Quantizer is given by a set of codevectors that form a codebook $\mathbf{Y}=\{\mathbf{y}_1,\ldots, \mathbf{y}_c\}$ (c is the codebook size), the encoding operation $\mathfrak{E}:\Re^d \rightarrow \{1,..,c\}$ that maps each vector in $\Re^d$ to the nearest codevector in the sense of some defined distance (most usually the Euclidean distance); and the decoding operation $\mathfrak{E}^{-1}$: $\{1,..,c\} \rightarrow \Re^d$ that reconstructs the encoded signal using the codebook. The Vector Quantization design consists of the estimation of the codebook from a sample $\mathfrak{X}=\{\mathbf{x}_1,\ldots, \mathbf{x}_n\}$.

The Bayesian approach to image processing [5, 2] is one of the more versatile approaches. The observed image is the degraded image **G** and **F** is the original or desired image. The *a posteriori* conditional density given by Bayes' rule

$$p\left(\mathbf{F}=\mathbf{f}|\mathbf{G}=\mathbf{g}\right) = \frac{p\left(\mathbf{G}=\mathbf{g}|\mathbf{F}=\mathbf{f}\right)p(\mathbf{F}=\mathbf{f})}{p(\mathbf{G}=\mathbf{g})} \tag{16}$$

has been used to find different types of estimates of the desired image **F** from the observed image **G**. The *maximum a posteriori* (MAP) and *maximum likelihood* (ML) are the modes of $p(\mathbf{F} = \mathbf{f} \mid \mathbf{G} = \mathbf{g})$ and $p(\mathbf{G} = \mathbf{g} \mid \mathbf{F} = \mathbf{f})$, respectively. In general, it is difficult to obtain the marginal density $p(\mathbf{G} = \mathbf{g})$, however, the MAP and ML estimates do not require it. Both estimation methods need to postulate models for the prior $p(\mathbf{F} = \mathbf{f})$ and conditional $p(\mathbf{G} = \mathbf{g} \mid \mathbf{F} = \mathbf{f})$ probabilities. The prior model specifies a broad class of images through the specification of

probabilistic relationships. The conditional probabilities can be postulated as a model of the image degradations or of the transformation between the observed and desired image.

In the VQ Bayesian filter (VQBF) the codevectors are considered to be centered around their middle pixels, like the conventional definition of convolution masks They are, thus redefined as $\mathbf{y}_i = \left(y^i_{k,l} : -\sqrt{d}/2 \le k,l \le \sqrt{d}/2\right)$. The image is not decomposed into blocks, rather we consider for each pixel a neighborhood $\mathbf{F}_{i,j} = \left\{F_{i+k,j+l} : -\sqrt{d}/2 \le k,l \le \sqrt{d}/2\right\}$. The filtering process corresponds to exchange each pixel by the central pixel in the codevector that encodes its neighborhood, denoting $\tilde{\mathbf{F}} = \left[\tilde{F}_{i,j} : 1 \le i,j \le N\right]$ the filtered image it can be formalized as follows: $\tilde{F}_{i,j} = \mathbf{y}_{0,0}^{\varepsilon(\mathbf{F}_{i,j})}$. The codevectors become the probabilistic models of the pixel neighborhood. To put the VQBF in the framework of Bayesian image processing, the filtering application of the codebook must be interpreted as realizing the following approximation of the posterior probabilities

$$p\left(\mathbf{F}_{i,j} = \mathbf{y}^k_{0,0} \middle| \mathbf{G}_{i,j} = \mathbf{g}\right) = \delta_{k,\varepsilon(\mathbf{g})} \quad 1 \le i,j \le N \text{ and } 1 \le k \le c \tag{17}$$

We recall the probabilistic model embodied by the codebook. In our works we do consider that the codebook design by the Evolutive Strategy intends to minimize the Euclidean distortion. The method employed is not relevant now as long as it minimizes

$$D = \sum_{i=1}^{n} \left\| \mathbf{x}_i - \mathbf{y}_{\varepsilon(\mathbf{x}_i)} \right\|^2 \tag{18}$$

A well-known interpretation [23], in terms of statistical decision theory, of the minimization of the Euclidean distortion beside is as follows: given a number of classes, i.e. *c*, and feature vectors whose probability density follows a mixture of conditional densities $p(\mathbf{x}) = \sum_{j=1}^{c} p\left(\omega_j\right) p\left(\mathbf{x} \middle| \omega_j\right)$. If we assume that the conditional densities are Gaussian with identical unit covariance matrices $p\left(\mathbf{x} \middle| \omega_j\right) \approx \mathbf{N}\left(\mathbf{y}_j, \mathbf{I}\right)$, and that the classes are equiprobable, then the minimization of the Euclidean distortion is equivalent to maximum log-likelihood estimation of the parameters of the model, the class means. Based on these parameters the MAP decision $\max_j p\left(\omega_j \middle| \mathbf{x}\right)$ is the Bayesian minimum risk decision. Thus, the filtering realized by VQBF corresponds to a MAP image process, in which the classes are the gray levels of the central pixel in the representative neighborhoods extracted from the image. . We can state the model of the dependencies of each pixel to its neighborhood as:

$$p\left(F_{ij} = f_{0,0} \middle| \mathbf{F}_{i,j} = \mathbf{f}\right) = \sum_{j=1}^{c} \frac{1}{c} \frac{1}{(2\pi)^{d/2}} e^{-\frac{1}{2}\left\|\mathbf{f}-\mathbf{y}_j\right\|^2} \tag{19}$$

Where $f_{0,0}$ is the central pixel of an image block $\mathbf{f}$.

The case for Bayesian analysis of VQBF properties is that of deducing from the statement of the posterior probabilities (19) the prior $p(\mathbf{F} = \mathbf{f})$ and conditional $p(\mathbf{G} = \mathbf{g} \mid \mathbf{F} = \mathbf{f})$ models. This analysis will support the VQBF approach and will be dealt with elsewhere.

4. - EXPERIMENTS AND RESULTS.

In this section we present the visual results of the application of the VQBF based on the codebooks computed by our Evolution-based Adaptation Strategy over an image. The sample size was the 10% of the original image. When we work with a of fixed number of classes the number of generations is set to 30 and the number of classes is set to $\mu = c = 16$. When we work with the selection that tries to determine the optimal number of classes the number of generations is set to 20 and the initial number of classes is the same as before. Other parameters for the evolution strategy are set to $T = 0.03$ (selection threshold) and $\eta = 0.5$ (more priority to local distortion than to entropy increment).

As the end interest of these images is for medical-biological inspection, the visual evaluation is the prime concern. Therefore, we present the visual results of the application of VQBF and several conventional approaches: the Median filter, the Gaussian smoothing, Gray level Morphological filters and the Wiener filter with noise self-estimation. To highlight the differences of the different filtering approaches, we show in the figure the equalization of the images after filtering. In all the cases we have considered neighborhoods of size 3, 5 and 7.

The approach has been tested over a Magnetic Resonance image obtained by the Unit of Magnetic Resonance of the Universidad Complutense. The original image is of 718x717 pixels and is shown in fig. 2a. The objective of the work is to enhance the image with some denoising algorithm and to detect the infected region enclosed by a white square in fig. 2a. The processing of the image must therefore, eliminate the Gaussian noise while preserving most of the structure of the image, specially in the interest region. To appreciate the denoising effects of the algorithms we perform the equalization of the original image (fig. 2b) and of the images after filtering figs. 3 to 9. To give a better impression of the preservation of the image structure we show the borders detected by applying a Laplacian operator on the filtered image and thresholding it.

The usual method to process noisy images before segmentation, when there is no known model of the distortion and the noise, is the application of smoothing filters. The results obtained by

the application of the median filter, Gaussian smoothing, gray level morphological filters and the Wiener filter appear, respectively, in figures 3 to 7. The results of our VQBF with fixed number of classes are shown in figure 8 and with variable number of classes in figure 9. Figures 10 to 16 shows the borders detected in the filtered images.

If we consider the results in terms of denoising and region segmentation, the general effect of conventional smoothing filters (Gaussian, Morphological, Median and Wiener) is a diffusion that distorts the region definition, blurring its boundaries. This negative effect increases as the size of the kernel increases. However, the VQBF shows a good denoising response while preserving the region definitions. Focussing into the infected region highlighted in fig. 2a, it can be appreciated that it is heavily blurred in figs. 3 to 7 while it is well preserved in figs. 8 and 9. Besides, VQBF shows no degradation by over blurring as the kernel size grows. The effect of our strategy to determine the optimal number of clusters can be appreciated comparing figs. 8 and 9, with figs. 16 and 17. Although the number of classes found is greater than in the constant size case, the visual results show some improvement, especially in the images of the detected borders. The search for the optimal number of classes produces the disappearance of the darker tissues in the image. However the infected region is well preserved in all the circumstances.

Focusing on the borders detected before and after filtering the image, the excellent properties of the VQBF are more clearly exhibited. Fig. 10 shows the borders detected in the original image. The smoothing decreases the magnitude of the detected borders, displaces and diffuses them. The extreme bad result is for the Gaussian smoothing with kernel 7x7 whose detected borders have a very small magnitude that needs a very low threshold. Conventional filters either loose the interest region or preserve many noisy borders. As the codevector dimension increases, the VQBF border detection improves. The detection of the optimal number of classes gives the best results in terms of isolating interesting regions. Both instances of the 7x7 VQBF preserve the main boundaries, especially in the interest region.

5.- CONCLUSIONS.

We have proposed the application of an Evolution-based Adaptation Strategy to estimate a vector quantizer with fixed and variable number of classes that is applied as filtering mechanisms. We have shown the results of those approaches against the results obtained by other conventional filtering and smoothing techniques widely used for noise removal. Ours approaches do not blur the image as the neighborhood size increases, and at the same time the

noise is removed more efficiently as the neighborhood size increases. This is more evident in the parts of the image that correspond to empty space, where our approach removes almost all the noise. The border detection also shown that VQBF defined regions of interest with accuracy.

Future work will be addressed to the improved optimal determination of the number of classes, that already gives very promising results.

Acknowledgments

Work partially supported by Departments of Education, and Industry of the Gobierno Vasco under project PI98-21; UE-1999-1 and the EC project "Polarized helium for imaging of the lung".

REFERENCES.

[1] A. Gersho, R.M. Gray, Vector Quantization and signal compression, Kluwer Acad. Pub., 1992.

[2] A.K. Jain, Fundamentals of digital image processing, Englewood-Clliffs: Prentice-Hall, 1989.

[3] M.S. Kankanhalli, B.M. Mehtre, J.K. Wu, Cluster based color matching for image retrieval, Pattern Recognition (1996) 29(4), 701--708.

[4] P.C Cosman, K.L. Oehler, E.A. Riskin, R.M. Gray, Using Vector Quantization for Image Processing, IEEE Proceedings (1993) 81(4) 1326—1341.

[5] S. Geman, D. Geman, Stochastic relaxation, Gibbs Distributions and the Bayesian Restoration of Images, IEEE trans. PAMI, 6(1984) 721—741.

[6] G.Winkler, Image analysis, random fields and dynamic Monte Carlo methods, Berlin Springer-Verlag, 1995.

[7] Alippi C., Cucchiara R., Cluster partitioning in image analysis classification: a genetic algorithm approach, Proc. IEEE COMPEURO, The Hague, (1992) 139--144.

[8] Andrey P., Tarroux P., Unsupervised image segmentation using a distributed genetic algorithm, Pattern Recognition, (1994) 27(5), 659--673.

[9] Babu G.P., Murty N.M., Clustering with evolution strategies. Pattern Recognition (1994) 27(2), 321--329.

[10] Bezdek J.C., Boggavaparu S, Hall L.O., Bensaid A., Genetic Algorithm guided Clustering., Proc. 1st IEEE Conf. Evol. Comp., (1994) 34--39.

[11] Bhuyan J.N., Raghavan V.V., Elayavalli V.K., Genetic algorithm for clustering with an ordered representation. Proc. 4th Int. Cong. Gen. Alg., (1991) 408--415.

[12] Blekas K., Stafylopatis A., Real coded genetic optimization of fuzzy clustering. EUFIT-96 (1996), Aachen, Germany.

[13] Buckles B.P., Petry F.E., Prabhu D., George R., Srikanth R., Fuzzy clustering with genetic search. Proc. 1st IEEE Conf. Evol. Comp., (1994) 46--50.

[14] Jones D.R., Beltrano M.A., Clustering with genetic algorithms. Operating Sciences Dept., General Motors Res. Lab., Warren, Mich. (1990)

[15] Ketaff F.Z., Asselin de Beauville J.P., Genetic and fuzzy based clustering. Proc. 5th Conf. Int. Fed. Class. Soc. (Kobe,Japan) (1996) , 100--103.

[16] Lucasius C.B., Dane A.D., Kateman G., On k-medoid clustering of large data sets with the aid of a genetic algorithm: background, feasibility and comparison. Analytica Chimica Acta 282, (1993), 647--669.

[17] Luchian S., Luchian H., Petriuc M., Evolutionary Automated Classification. Proc 1st IEEE Conf. Evol. Comp., (1994) 585--588.

[18] Moraczewski I.R., Borkowski W., Kierzek A., Clustering geobotanical data with the use of a genetic algorithm. COENOSES (1995), 10(1), 17--28.

[19] S. M. Bhandarkar, H. Zhang, Image Segmentation Using Evolutionary Computation, IEEE Transactions on Evolutionary Computation, (1999) 3(1) 1-21

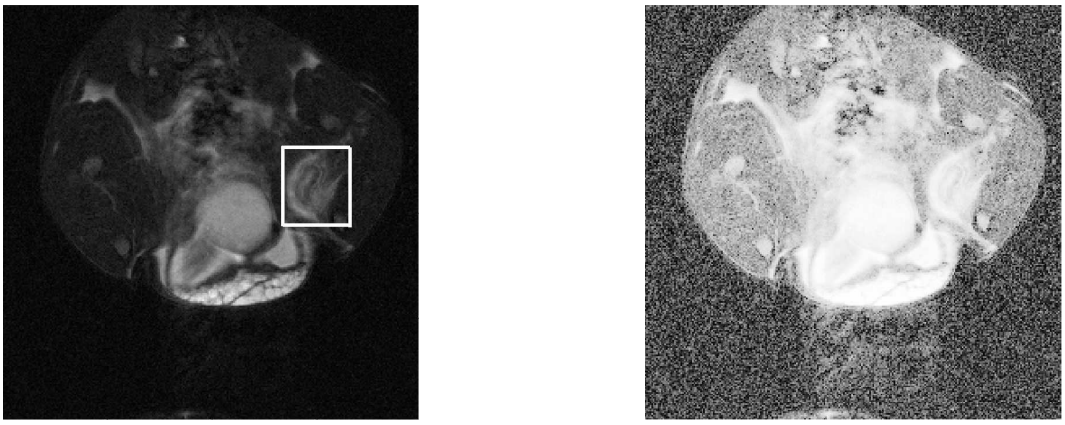
[20] L. O. Hall, I. B. Özyurt, and J. C. Bezdek, Clustering with a Genetically Optimized Approach, IEEE Transactions on Evolutionary Computation, (1999) 3(2) 103-112

[21] A.I. González, M. Graña, F.X. Albizuri, A. D'Anjou, F.J. Torrealdea, A near real-time Evolution-based Adaptation Strategy for dynamic Color Quantization of image sequences, Information Sciences 122 (2000), 161—183.

[22] T. Bäck, H.P. Schwefel, An overview of Evolutionary Algorithms for parameter optimization, Evolutionary Computation 1(1993) 1--24.

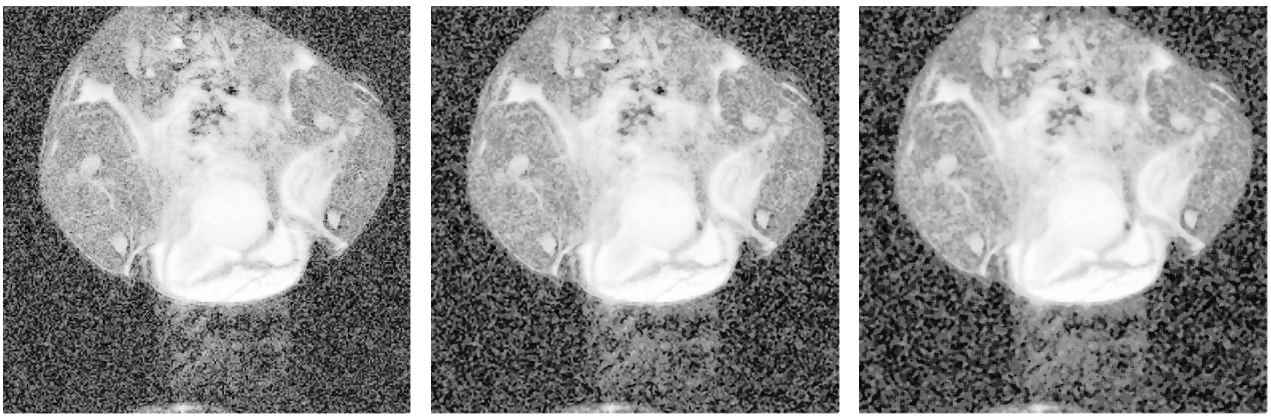
[23] R.D. Duda, P.E. Hart, Pattern Classification and Scene Analysis, Wiley, 1973.

[24] Z. Michalewicz, Evolutionary computation: practical issues, IEEE ICEC'96, (1996), 30—39.



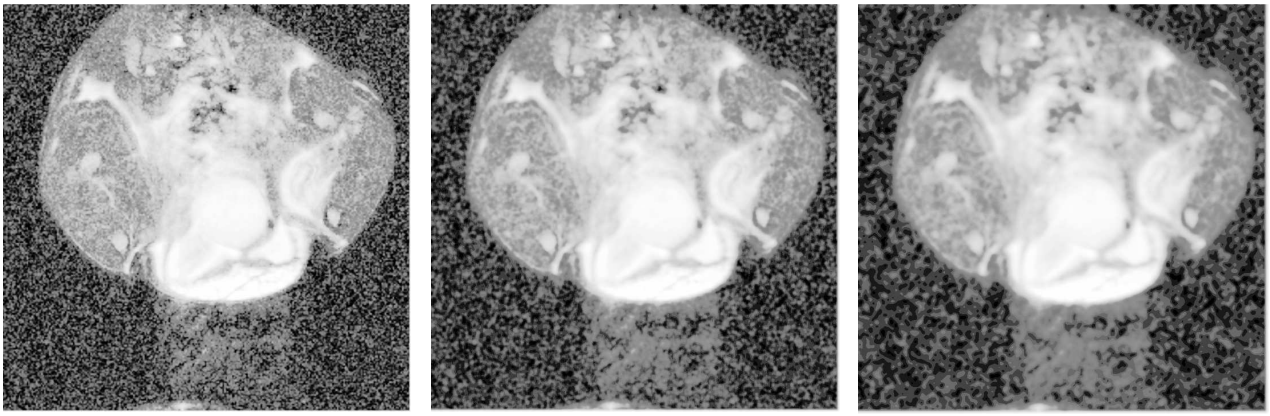
(a) (b)

Figure 2. Original image (a). After equalization (b).



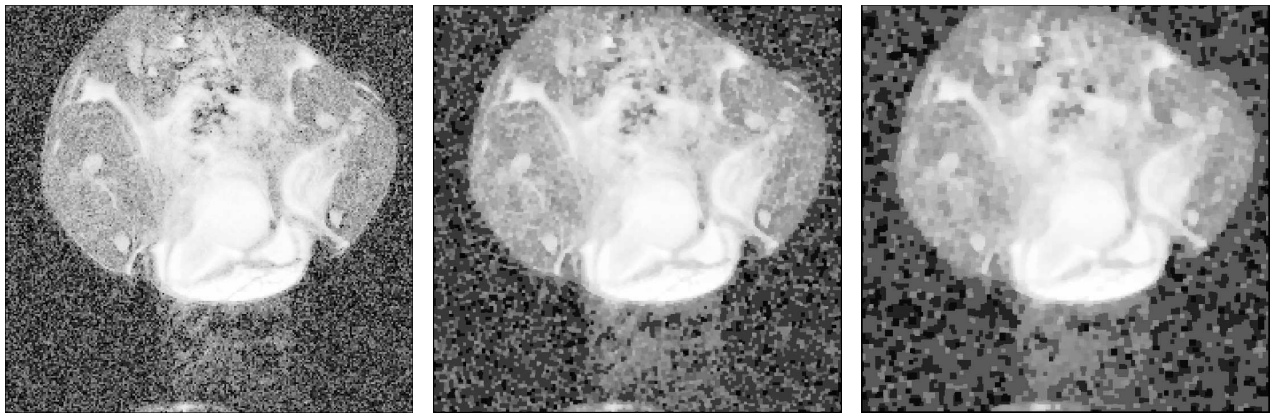
3x3 5x5 7x7

Figure 3. The equalization of images filtered with Median Filter method and several neighborhood/radius sizes.



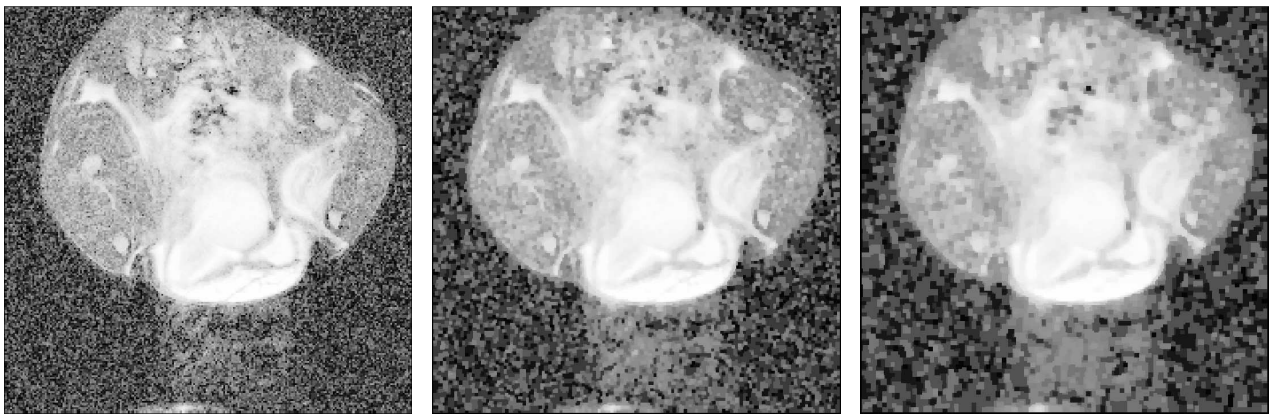
3x3 5x5 7x7

Figure 4. The equalization of images filtered with Gaussian Filter method and several neighborhood/radius sizes.



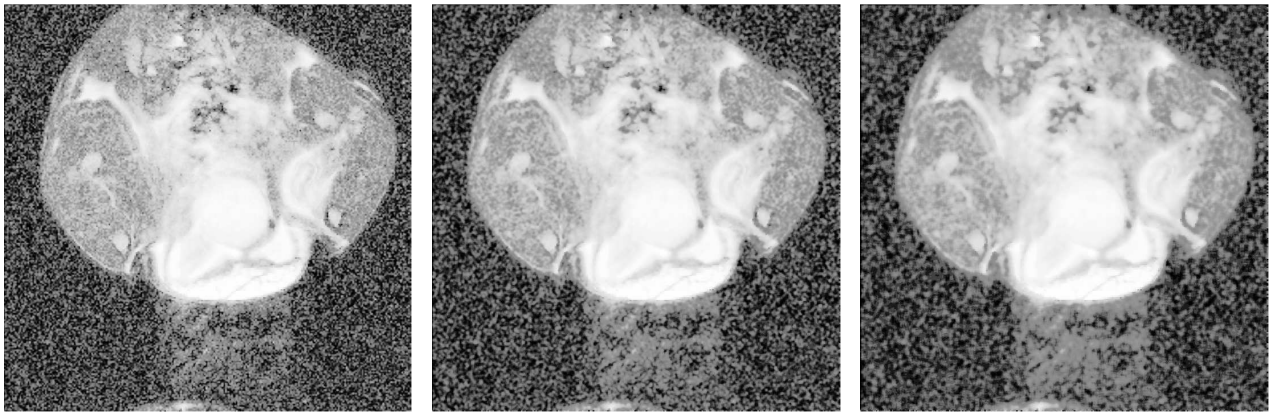
3x3 5x5 7x7

Figure 5. The equalization of images filtered with Opening+Closing Morphological Filter method and several neighborhood/radius sizes.



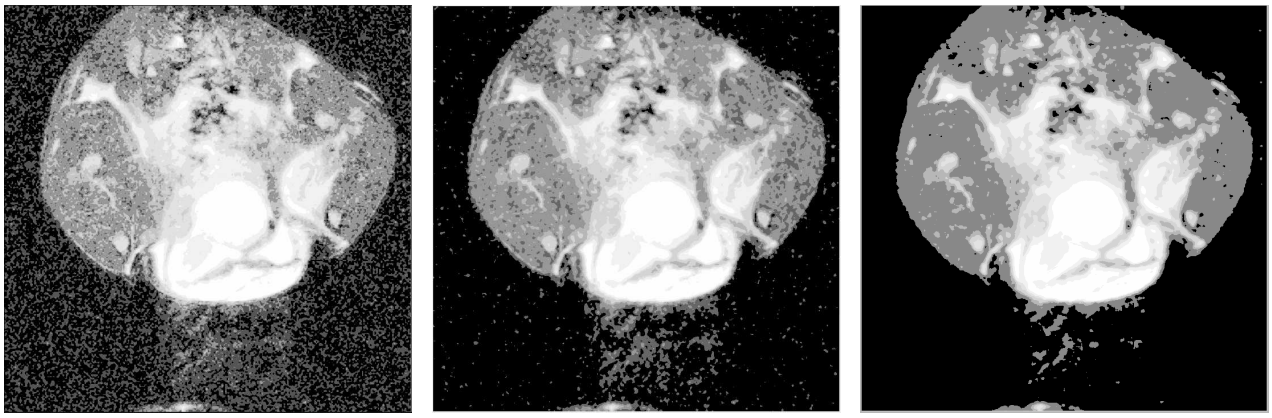
3x3 5x5 7x7

Figure 6. The equalization of images filtered with Closing+Opening Morphological Filter method and several neighborhood/radius sizes.



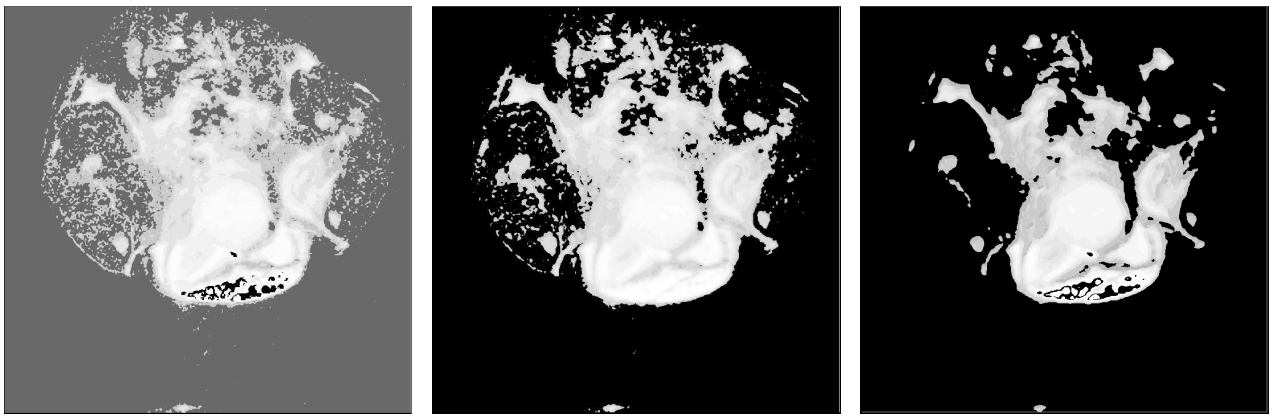
3x3 5x5 7x7

Figure 7. The equalization of images filtered with Wiener Filter method and several neighborhood/radius sizes.



3x3 5x5 7x7

Figure 8. The equalization of images filtered with ES + VQBF method and several neighborhood/radius sizes. The number of classes is constant (c=16).



3x3 5x5 7x7

Figure 9. The equalization of images filtered with ES + VQBF method and several neighborhood / radius sizes. The number of classes is determined automatically by the Evolution Strategy (number of classes obtained from left to right: 24, 21, and 22).

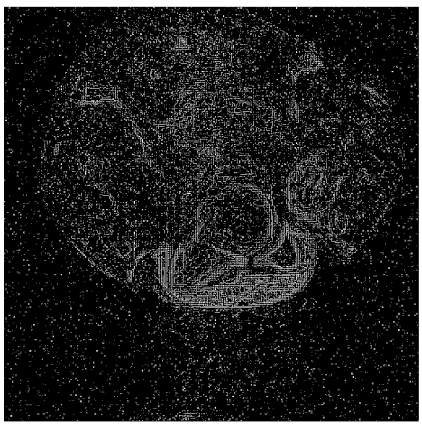
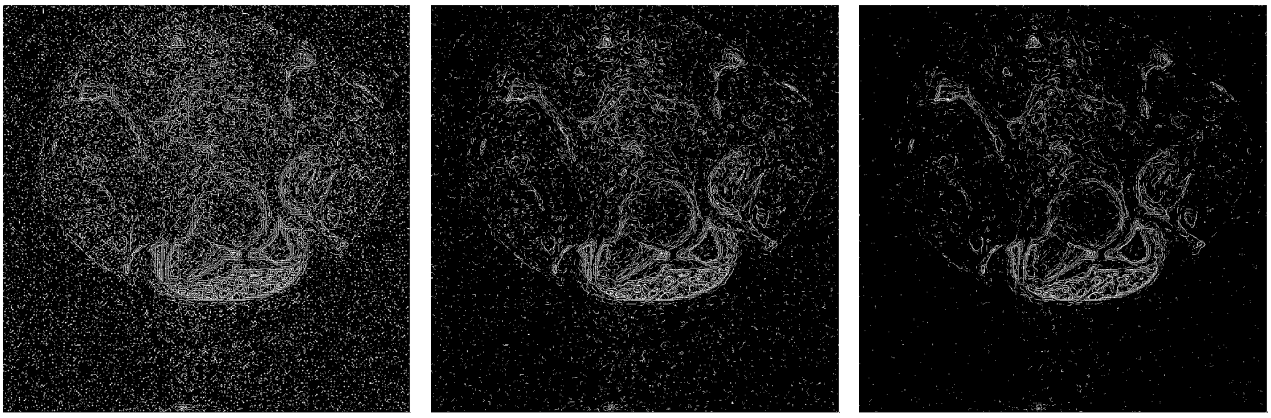


Figure 10. Borders detected in the original image (threshold 32).

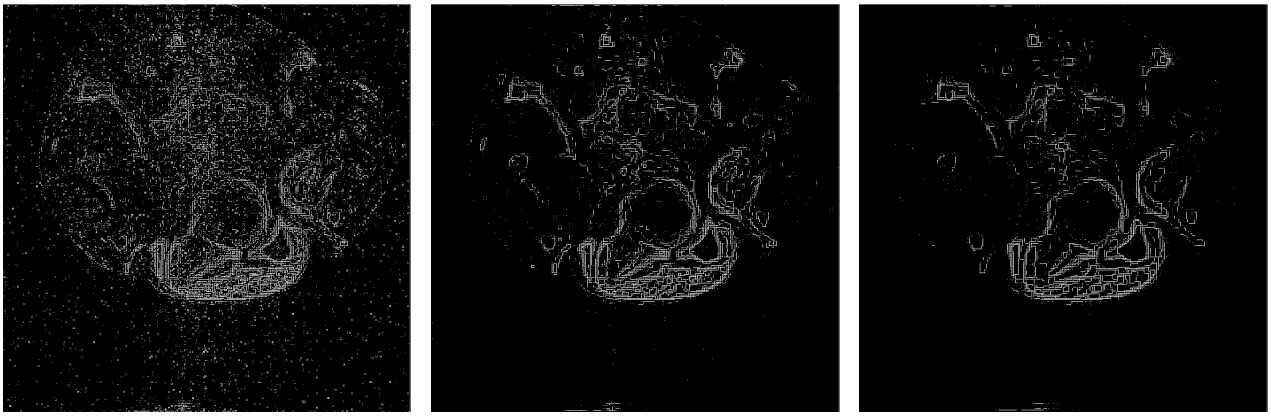


3x3 5x5 7x7

Figure 11. Borders of images filtered with Median Filter method and several neighborhood/radius sizes (threshold 16).

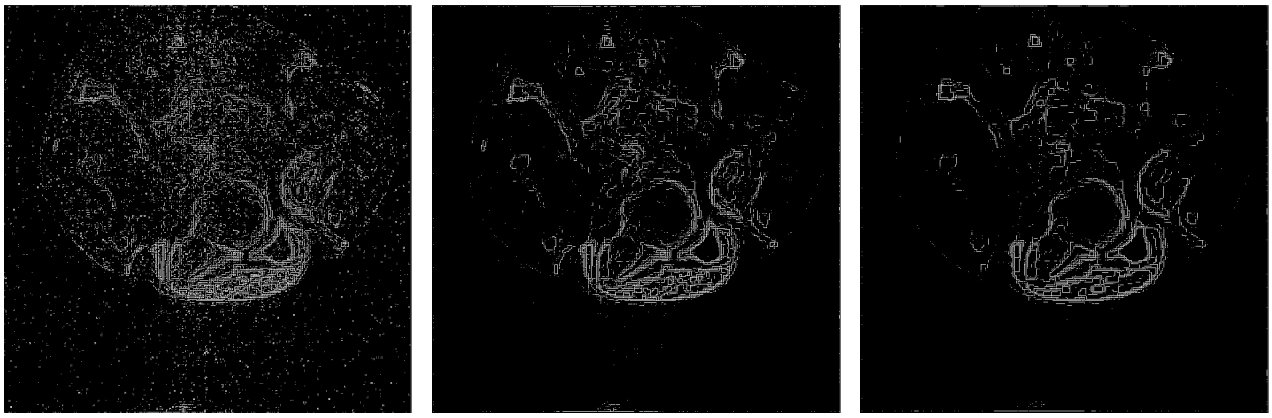
3x3 5x5 7x7

Figure 12. Borders detected in the images filtered with Gaussian Filter method and several neighborhood/radius sizes (threshold 12).



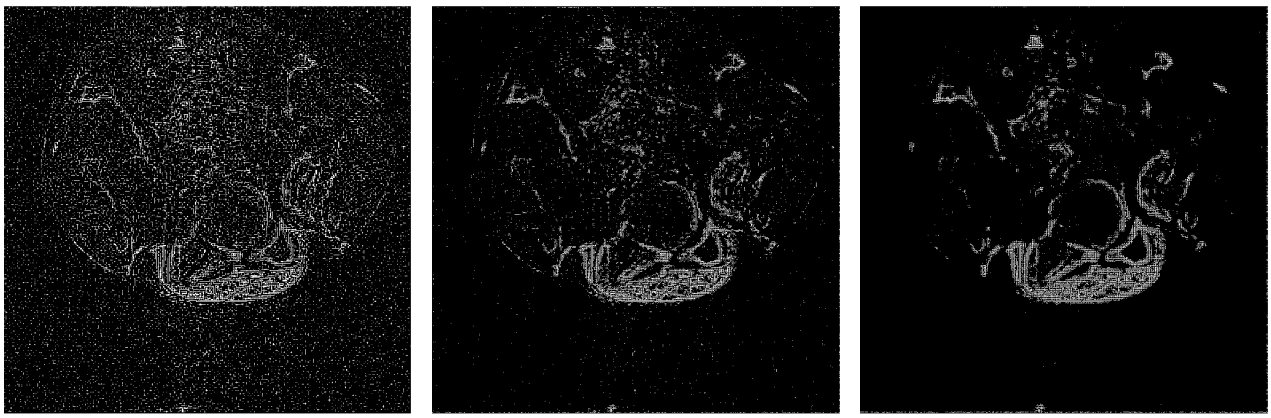
3x3 5x5 7x7

Figure 13. Borders detected in the images filtered with Opening+Closing Morphological Filter method and several neighborhood/radius sizes (threshold 32).



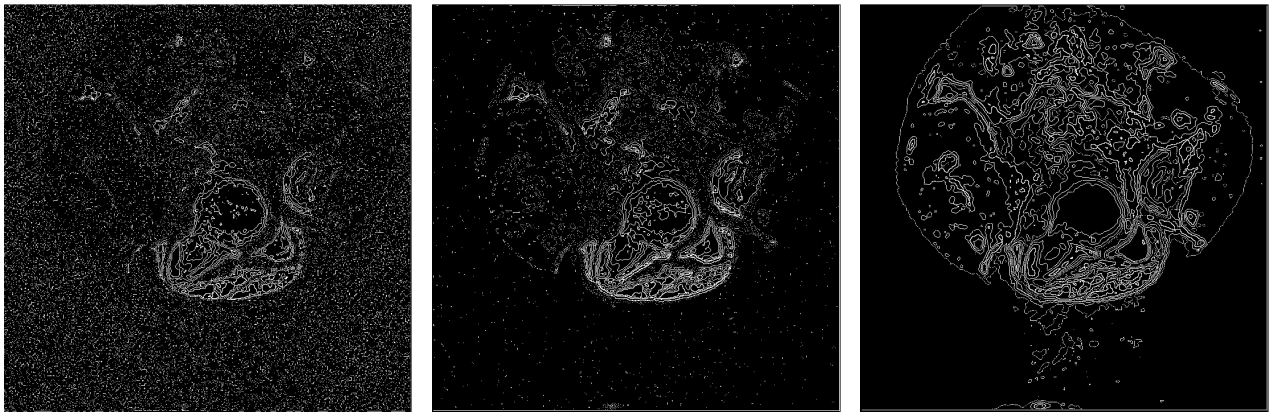
3x3 5x5 7x7

Figure 14. Borders detected in the images filtered with Closing+Opening Morphological Filter method and several neighborhood/radius sizes (threshold 32).



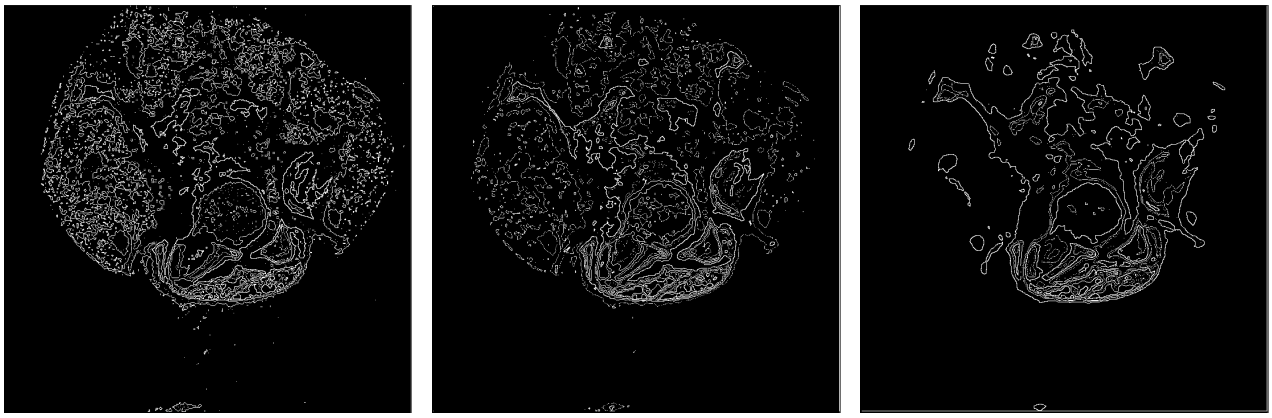
3x3 5x5 7x7

Figure 15. Borders detected in the images filtered with Wienner Filter method and several neighborhood/radius sizes (threshold 12).



3x3 5x5 7x7

Figure 16. Borders detected in the images filtered with ES + VQBF method and several neighborhood/radius sizes. The number of classes is constant *c*=16 (threshold 32).



3x3 5x5 7x7

Figure 17. Borders detected in the images filtered with ES + VQBF method and several neighborhood / radius sizes. The number of classes determined by the Evolution Strategy (threshold 32).