Evolutionary Design of Robust Noise-Specific Image Filters

Zdenek Vasicek
Brno University of Technology
Faculty of Information Technology
Božetěchova 2, 61266 Brno, Czech Republic
Email: vasicek@fit.vutbr.cz

Michal Bidlo
Brno University of Technology
Faculty of Information Technology
Božetěchova 2, 61266 Brno, Czech Republic
Email: bidlom@fit.vutbr.cz

***Abstract*—Evolutionary design has shown as a powerful technique in solving various engineering problems. One of the areas in which this approach succeeds is digital image processing. Image filtering represents a wide topic in 2D signal processing. In this case different types of noise are considered in the filtering process to restore the image quality that has been decreased by changing values of some pixels in the image (e.g. due to the transmission through unreliable lines or in the process of acquiring the image). Impulse noise represents a basic type of non-linear noise typically affecting a single pixel in different regions of the image. In order to eliminate this type noise median filters have usually been applied. However, for higher noise intensity or wide range of the noise values this approach leads to corrupting non-noise pixels as well which results in images that are smudged or lose some details after the filtering process. Therefore, advanced filtering techniques have been developed including a concept of noise detection or iterative filtering algorithms. In case of the high noise intensity, a single filtering step is insufficient to eliminate the noise and obtain a reasonable quality of the filtered image. Therefore, iterative filters have been introduced. In this paper we apply an evolutionary algorithm combined with Cartesian Genetic Programing representation to design image filters for the impulse noise that are able to compete with some of the best conventionally used iterative filters. We consider the concept of noise detection to be designed together with the filter itself by means of the evolutionary algorithm. Finally, it will be shown that if the evolved filter is applied iteratively on the filtered image, a high-quality results can be obtained utilizing lower computational effort of the filtering process in comparison with the conventional iterative filters.**

***Index Terms*—evolutionary design; image filter; impulse noise**

I. Introduction

Linear filters became the most popular image signal processing systems in the past years. The reason of their popularity is caused by the existence of robust mathematical models which can be used for their analysis and design. However, there exist many areas in which the *nonlinear filters* provide significantly better results. The advantage of nonlinear filters lies in their ability to preserve edges and suppress the noise without loss of details. The success of nonlinear filters is caused by the fact that image signals as well as existing noise types are usually nonlinear.

The impulse noise is the most frequently referred type of noise. While it is very difficult to remove the impulse noise using linear filters as they tend to smudge the resulting images, the nonlinear filters are able to provide filtered images with a reasonably good quality.

In most cases, the impulse noise is caused by malfunctioning pixels in camera sensors, faulty memory locations in hardware, or errors in the data transmission. We distinguish two common types of impulse noise: *salt&pepper noise* (commonly referred as intensity spikes) and *random-valued shot noise*. For images corrupted by the salt&pepper noise, the noisy pixels can take only the maximum or minimum values. In case of grayscale images, the noisy pixels are represented by the value 255 or 0 (no other values are considered). However, this kind of noise is a theoretical (model) instance rather than a corruption frequently occurred in praxis. In case of the random-valued impulse noise, the noisy pixels can have an arbitrary value (from 0 to 255 in case of grayscale images) uniformly distributed in this range. It is evident that this type of noise is more difficult to remove in comparison with salt&pepper noise because the noisy pixels may actually possess any value representing the uncorrupted pixels. Therefore, the noisy pixels are difficult to detect which represents a problem especially in the images containing many details because the filtering process may tend to remove the details assuming their pixels being corrupted by the noise. On the other hand, if the noise occurs in a smooth grade transition, where the difference between the noise pixel and the neighbouring pixels is low, the noise may not be recognized and remains in the filtered image which degrades its quality. A serious problem represents the case of random-valued noise if occurred in a high intensity in which the common single-step filters do not usually provide a reasonable quality. Thus the iterative filtering may be the solution.

The *common median filter* (MF) [1], initially introduced to eliminate impulse noise, continues to be the most used approach even if the quality of the filtered images is poor in comparison with the other advanced techniques. The main advantage of the common median filter lies in its simple and effective software as well as hardware implementation which is straightforward and does not require many resources. The median-based filtering approach has been intensively studied and extended to promising approaches such as *Center Weighted Median Filter* (CWMF) [2], more general *Weighted Median Filter* (WMF) [3] or *weighted order statistic filter*

[4]. Nevertheless, all these median-based methods tend to smudge the image since applying the median filtering to the entire image would inevitably remove details presented in the image. In order to overcome this drawback, a *switching-based median filtering* concept has been proposed [5]. This concept splits the filtering process into two parts – noise detection and noise replacement. The noise detector determines which pixels are affected by the impulse noise and only these pixels are replaced (e.g. using MF). Noise detection can be based on various concepts: a median-based filter [5], fuzzy techniques [6] or neural networks [7]. The common problem of the before-mentioned detection mechanism is the necessity to predetermine the value of a threshold parameter which significantly influences the filtering quality. Therefore these approaches are rarely used in practice. The *adaptive median filter* (AMF) proposed in [8] is a robust approach which tries to identify and replace the affected pixels only. In contrast with the previous approaches, the detection method is based on the statistical ordered filters with gradually increasing kernel size [9]. In contrast with the median-based methods, AMF provides significantly better results even for the images corrupted with high intensity impulse noise. Apart from the non-iterative algorithms, the iterative algorithms such as *Pixel-Wise Median of the Absolute Deviations from the median* (PWMAD) [10] or *Directional Weighted Median filter* (DWM) outlined in [11] have been introduced. These approaches provide very good results if the random valued impulse noise is considered, do not contain any varying parameters and require no previous training or optimization. The main disadvantage is apparent – the iterative approach place higher requirements to the memory resources especially in case of hardware implementation.

Two different approaches can be identified to perform image filtering using evolutionary techniques: (1) An existing filter is considered for a given noise type and the evolution is applied to find suitable coefficients of the filter (i.e. it is a case of filter optimization). An instance of this approach for the filtering of color images can be found in [12]. (2) The evolutionary algorithm is used to create entirely new filter considering a set of suitable functions and limitations specified by human designer (it is a case of evolutionary design of a new filtering algorithm) [13]. In this paper we focus on the second approach.

The goal of this paper is to design robust impulse-noise filters by means of Cartesian Genetic Programming and to show that the evolutionary algorithm is able to design filters of at least the same or better performance in comparison with the conventional solutions. The proposed results will be compared with the conventional median filters (common MF and AMF) and the iterative filters DWM and PWMAD that provide very good results in filtering images corrupted by a higher noise intensity. We show that the filters designed by the evolution are able to compete with these solutions from the point of view of the filtering quality with respect to the computational effort (i.e. the number of iterations) that is needed to filter an image. Two sets of experiments will be presented considering salt&pepper noise and random-valued impulse noise.

II. Switching based filtering concept

The main disadvantage of the common median-based filters is that the filtering transformation is applied on all the pixels of the image regardless of the pixel represents the noise or not. Thus this approach results in the loss of the image details and causes the degradation of the image quality especially if a larger filter kernel is used. In order to improve the filtering quality, the switching-based median filter has been proposed. The switching-based approach outlined in [5] can be considered as a general process of filtering that operates in two steps. In the first step, the noisy pixels are detected using a detection algorithm. Then, the new values of the corrupted pixels are estimated using an estimation algorithm.

Let x_{ij} and y_{ij} denote pixels with coordinates i, j in a noisy and a filtered image respectively. If the estimated value of the corrupted pixel x_{ij} is z_{ij}, the switching filter concept can be defined as

$$y_{ij} = s_{ij} \cdot z_{ij} + (1 - s_{ij}) \cdot x_{ij}$$

where s_{ij} is a binary noise map – an output produced by the estimation algorithm. Noise map s_{ij} contains ones at the positions of pixels detected as noisy pixels.

In general, s_{ij} is determined by comparing the absolute difference between the original pixel value x_{ij} and some local statistics $\Omega(x_{ij})$ with a threshold T. Statistics $\Omega(x_{ij})$ can be produced by common median filter, weighted median filter, adaptive median filter or using a complex detection mechanism, e.g. DWM or PWMAD. Since the value of T is highly correlated to the image contents, noise probability and distribution, T has to be calculated for each filtered image. This is unpractical since the problem of finding the optimal threshold is a complex task. While setting T too high leaves a lot of the noisy pixels unfiltered, too low T causes that image details will be treated as noise and the overall image quality will be degraded. In order to avoid setting of this parameter, the process of noise map estimation is usually applied iteratively with varying threshold (e.g. DWM). Estimated value of the filtered pixel z_{ij} is usually based on common median filter or its variants (e.g. weighted median filter).

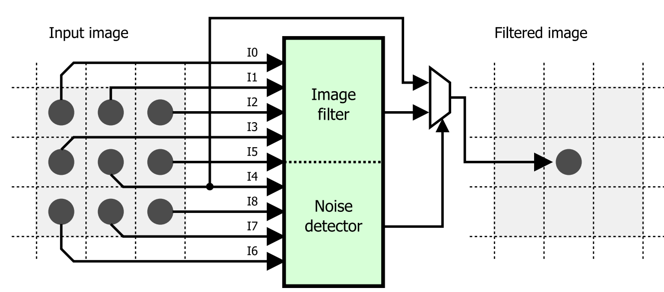


Fig. 1. The concept of the switching-based filtering using a 3×3 filter kernel

The concept the switching-based filtering is shown in Figure 1. In this paper the task is to design a circuit with two 8-bit outputs; the first output represents the image filter output and the second one is the output of the noise detector. If the most significant bit of the noise detector output equals 1, then the filtered value is taken from the image filter (i.e. the processed pixel was detected as noise), otherwise the original value is used.

III. Cartesian Genetic Programming

Cartesian Genetic Programming (CGP) has been introduced by J. F. Miller and P. Thompson in [14]. CGP was originally intended for the gate-level evolution, however, it has been extended for the functional level evolution and successfully applied in many areas. In its basic version, a candidate solution is directly represented in the chromosome which is represented by a sequence of integers.

A. Representation

In contrast with the common Genetic Programming, which encodes a candidate solution using a tree representation, CGP utilizes a matrix consisting of the fixed number of programmable nodes. More precisely, a candidate solution is modeled as an array of n_r (rows) $\times n_c$ (columns) of programmable nodes. Each programmable node can have several inputs n_i and outputs n_o and can implement one of the predefined functions; usually $n_i = 2$ and $n_o = 1$. The number of the primary inputs p_i, primary outputs p_o as well as the number of programmable nodes and their parameters are fixed. Each node input can be connected either to the output of a node placed in the previous l columns or to one of the primary inputs. The parameter called as l-back, in fact, defines the level of connectivity and thus reduces or extends the search space. For example, if $l = 1$ only neighboring columns may be connected; if $l = n_c$, full connectivity is enabled. Feedback is not allowed in the basic version of CGP; in order to avoid a feedback, the inputs are not allowed to connect to elements in the same and any consequent column. The primary outputs can be connected usually to the output of any programmable node since the direct connection to the primary inputs implements the identity function only. Each programmable node can perform one of the f_i-input functions predefined in a set Γ ($0 \leq f_i \leq n_i$). The Γ is chosen according to the application requirements; while in case of the gate-level evolution of digital circuits Γ consists of two-input boolean functions only, in case of functional level evolution Γ contains complex building blocks such as adders, multipliers etc.

A candidate solution is encoded using a sequence of integers that encodes the interconnection of the programmable nodes, their configuration (i.e. index of the function that a certain node implements) and the connection of the primary outputs. In order to encode a candidate solution, the primary inputs as well as the outputs of the programmable nodes have to be assigned a unique index. The primary inputs have assigned the indices $0, \ldots, p_i - 1$. Each output of each programmable node has assigned an integer starting by p_i and increasing by one and numbered row by row, column by column. Every n_i-input programmable node is fully encoded by a $e_i + 1$-tuple of integers. The first n_i integers encodes the connection of each

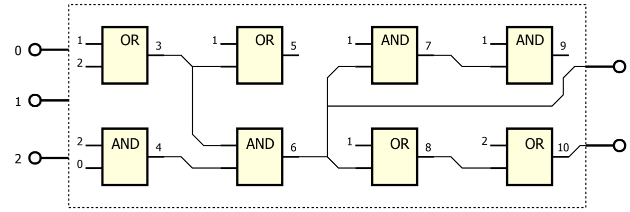


Fig. 2. An example of a candidate circuit in Cartesian Genetic Programming. The CGP parameters are as follows: $n_r = 2$, $n_c = 4$, $l = 3$, $n_i = 2$ (for all the nodes), $n_o = 1$ (for all the nodes), $p_i = 3$, $p_o = 2$, Γ = {AND (**0**), OR (**1**)}. Nodes 5,7 and 9 are not utilized. The corresponding chromosome is: 1,2,**1**, 2,0,**0**, 1,3,**1**, 3,4,**0** 1,6,**0**, 1,6,**1**, 1,7,**0**, 2,8,**1**, 6, 10. The last two integers indicate the outputs of the program. The function code of a gate is typed in bold.

node input. The last integer in the tuple encodes the function of the programmable node. The chromosome is encoded into a sequence of $(n_i + 1) \cdot n_c \cdot n_r + p_o$ integers. The last p_o integers encode the connection of the primary outputs.

Figure 2 shows an example of a candidate circuit and the corresponding chromosome in the CGP encoding. It can be seen that while the size of the chromosome is fixed, the size of phenotype is variable (i.e. some nodes may not be used).

B. Search Algorithm

CGP usually operates with a small population of $1 + \lambda$ individuals (typically, $\lambda = 4 - 20$). The initial population is generated randomly. Every new population consists of the best individual and its variants (mutants). The variants are created using a point mutation operator. In case when two or more individuals have received the same fitness score in the previous population, the individual which did not serve as a parent in the previous population will be selected as a new parent. This strategy is used to ensure the diversity of the population.

The fitness function usually takes one of two forms. For the symbolic regression problems, a training set is used. The goal is to minimize the difference between the output of a candidate program and the required output. For the evolution of logic circuits, all the possible input combinations are applied at the candidate circuit inputs, the outputs are collected and the goal is to minimize the difference between the obtained truth table and the required one. The evolution is stopped when the best fitness value stagnates or the maximum number of generations is exhausted.

IV. Image Filter Design Using CGP

Evolutionary design of image filters at the functional level has been introduced in [13]. In case of the evolutionary design of image filters we are addressing the following problem. Given an image corrupted by a certain type of noise that needs to be removed. Let us denote it as I_c. A reference image I_r is available that represents the uncorrupted version of I_c. The task is to construct a filter (using an evolutionary algorithm) working with a filter kernel $k \times k$ pixels that suppresses the given type of noise from I_c according to the knowledge contained in this image.

TABLE I
THE LIST OF FUNCTIONS THAT CAN BE IMPLEMENTED IN EACH PROGRAMMABLE NODE

code	function	description
0	255	constant
1	x	identity
2	$255 - x$	inversion
3	$max(x, y)$	maximum
4	$min(x, y)$	minimum
5	$x \gg 1$	division by 2
6	$x \gg 2$	division by 4
7	$x + y$	addition
8	$x +^S y$	addition with saturation
9	$(x + y) \gg 1$	average
10	y if $(x > 127)$ else x	conditional assignment
11	$\|x - y\|$	absolute difference
12	$x \ll 1$	multiplication by 2 with saturation
13	$x \ll 2$	multiplication by 4 with saturation

In order to evolve an image filter, the CGP at the functional level can be utilized since a particular image filter can be considered as a digital circuit having $p_i = k \cdot k$ 8-bit inputs and a single 8-bit output where k corresponds to the dimensions of the filter kernel (e.g. $p_i = 9$ in case of the 3×3 kernel).

The Fig. 2 shows the concept of evolutionary design of a image filter. Every pixel value of the filtered image is calculated using a corresponding pixel and its eight neighbors in the processed image (for the simplicity 3×3 kernel is considered).

In order to evolve an image filter which removes a given type of noise from a corrupted image, we need (a) a set of suitable functions (building blocks of the filter circuit) and (b) a training data to measure the fitness of the candidate filters (i.e. their quality). The goal of the evolutionary algorithm is to minimize the difference between the original image and the filtered image. The generality of the evolved filters (i.e., the ability to operate sufficiently also for other images containing the same type of noise the filters have not been trained for) is tested by means of a test set.

An image filter can be encoded using a CGP matrix consisting of $n_r \times n_c$ programmable nodes, $p_i = k \cdot k$ and $p_o = 1$. Note that the inputs and outputs of the nodes as well as the primary inputs and primary outputs ate considered as 8-bit ports. Each node can implement one of the predefined set of functions. For the purpose of the evolutionary design of non-linear filters we are using the functions listed in Table I. The utilized function set contains the standard non-linear operations such as minimum, maximum or absolute difference.

A. Evaluation of Candidate Filters

Every 8-bit pixel value of the filtered image is calculated using the value of the corresponding pixel in the corrupted image and its neighbouring pixels. Since it is impossible to evaluate all the input test vectors in the fitness function (for example, there are in total 256^{25} different combinations for the 5x5-pixel kernel in 256 degrees of grayscale), it is necessary to select a representative training data to calculate the quality of the candidate filters. Commonly the training data is created as a corruption of a reference image by applying a noise model to this image. The training data represents an important information for evaluating the results of candidate filters during the evolutionary process. The corrupted image provides training vectors that are applied as the input for the candidate filter. The output of the filter (i.e. the filtered image – I_f) is compared to the reference image in order to evaluate the filter quality. For the 5x5-pixel filter kernel the training vectors are generated as a set of 25-tuples (from the corrupted image) and the corresponding correct values (from the reference image).

The goal of the evolutionary algorithm is to design a filter which minimizes the difference between I_f and I_r. The evolved filter is required to be robust, i.e. it provides a good filtering quality not only on the training data but also on other real images for which the reference version is not known. Note that this ability can usually be achieved only in case of the type of noise the filter was trained for.

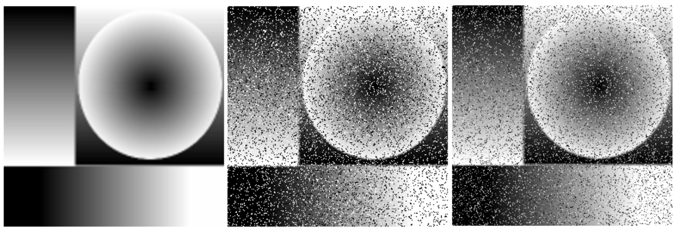
If the corrupted image is of the size $R \times C$ pixels and we are using a square-shaped input mask of 5x5-pixel filter kernel, then the number of filtered pixels is equal to the $(R-4) \times (C-4)$. The quality of the evolved image filter is expressed by the fitness function (the value of this function is to be minimized) shown in Eguation 1. In this equation, we consider that the pixels are indexed from 0 to $R - 1$ or $C - 1$.

$$fit(I_f, I_r) = \sum_{i=2}^{R-3} \sum_{j=2}^{C-3} \left| I_f(i,j) - I_r(i,j) \right| . \qquad (1)$$

B. Experimental Setup

The following setup of the CGP was utilized for the experiments presented in this paper. The CGP array consists of $n_c \times n_r$ = 7×9 nodes. The l-back parameter has been set to n_c (i.e. the full connectivity), however, only the elements situated in the first four columns can be connected directly to the primary inputs. The evolutionary algorithm works with the population of $\lambda = 8$ individuals. Up to 15 genes in an individual can be mutated. The initial population is generated randomly. The results were obtained from 100 independent runs of the CGP system. Each single experiment takes 200,000 generations. The goal of each evolutionary experiment is to find a filter working with 5x5-pixel kernel for the filtering the given type of noise.

The training data are provided by an artificial 256x256-pixel image corrupted by 20% noise which is illustrated in Figure 3. The training vectors are generated for the evolution of the salt&pepper noise and random-valued impulse noise filters. There are 62,061 unique training vectors for the salt&pepper noise in Fig. 3b and 63,437 unique training vectors for the random-valued impulse noise in Fig. 3c. The utilization of artificial image for generating the training data is in virtue of our experience in evolving filters for different types of noise. The image contains crucial features that should be preserved in the filtered image – smooth gradients of different types combined with sharp edges. These components showed as important for the filter to be trained on. The experiments showed that if the noise intensity is low during the training process ($\leq$ 10%), the training data does not provide a sufficient amount of information for the filter to be robust (i.e. in most cases the resulting filter is not able to remove noise of a higher intensity). On the other hand, if the training noise intensity is high ($\geq$ 30%), substantial amount of (training) image data is lost and the resulting filters do not exhibit reasonable filtering quality.



(a) (b) (c)

Fig. 3. The training data utilized in the experiments: (a) the reference image, (b) the image corrupted by 20% salt&pepper noise, (c) the image corrupted by 20% random-valued impulse noise. The noise intensity 20% means that value of 20% of pixels of the reference image is was changed (corrupted by the noise).

C. Principles of Conventional Iterative Filters

This section briefly summarizes the principles of the conventional iterative filters that were chosen for the comparison with our evolved solutions. The filters DWM and PWMAD were selected because they are considered as state-of-the-art nonlinear filters designed to suppress the impulse noise

The DWM filter has introduced an impulse detector which makes use of the difference between the current pixel and its neighbors aligned with four main directions. After the impulse detection, the filter does not simply replace the noisy pixels identified by the outputs of the median filter but continues to use the information of the four directions to weight the pixels in the window in order to preserve the details as removing the noise. A threshold, that is decreased in each iteration, is utilized to identify the noisy pixels. The DWM filter is supposed to perform much better than the other median-based filters in removing random-valued impulse noise, especially for higher noise intensity. Furthermore, it can preserve more detail features, even thin lines [11].

The PWMAD filter calculates (five-times iteratively and for each pixel in the image) the median and the absolute deviation of the median and the original pixel value. Then the median of the absolute deviations is calculated to determine the value of the filtered pixel.

V. Experimental Results

Two sets of experiments were performed. The first set considered the evolution of salt&pepper noise filters and the second set was devoted to the design of random-valued impulse noise filters. This section summarizes the obtained results and discusses their properties in comparison with some conventionally designed filters for each type of noise.

The quality of the evolved filters will be compared to several conventional single-step and iterative filters providing the best results in removing the impulse noise. In order to show the ability of the evolved solutions to improve the filtered image using the iterative processing, one and two iterations of these filters will be performed and the results compared to the images filtered by DWM and PWMAD. Moreover, the filtering results will also be compared to standard median filter (MF) and adaptive median filter (AMF). It will be shown that the evolved filters are able to provide the output quality that is very close to (or even better than) the results of the conventional filters and in less iterations in comparison with the conventional filters.

The resulting filters obtained by means of the CGP system were evaluated using a set of 30 different images, each of which was corrupted by 1%–30% noise. The filtering quality (expressed as the Peak Signal to Noise Ratio – PSNR) is calculated as the average of the PSNR for each image in the evaluation set and the noise intensity.

TABLE II
Comparison of the salt&pepper noise filters in terms of PSNR (dB). The size of the kernel is also specified for each filter.

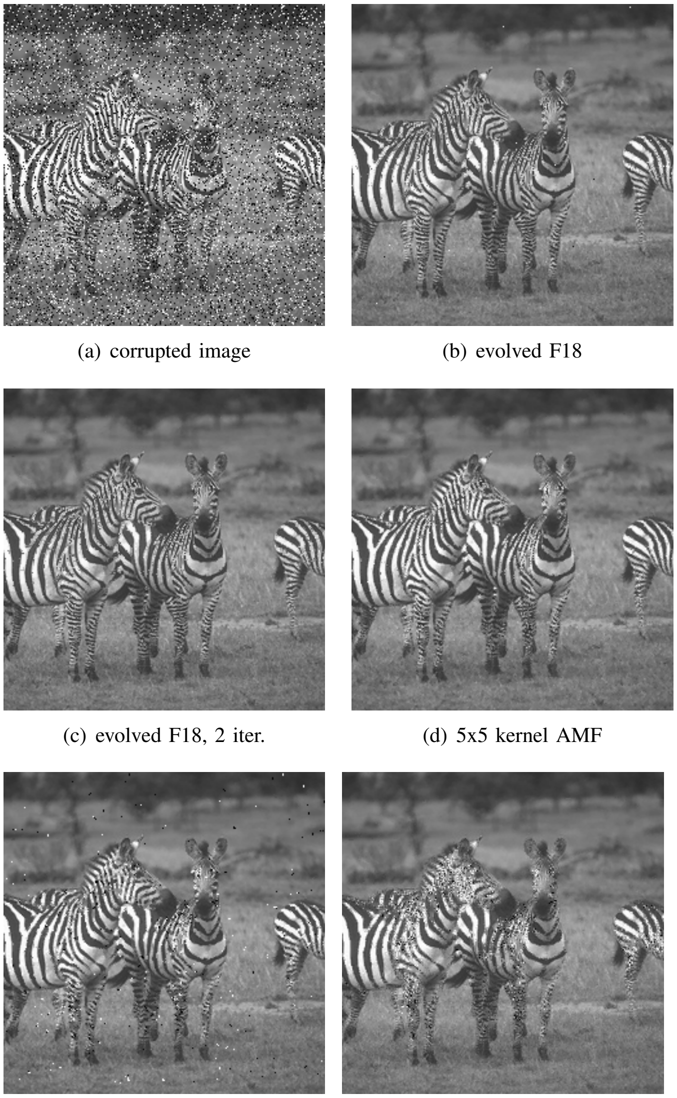
	noise intensity in percent						
filter	**1**	**5**	**10**	**15**	**20**	**25**	**30**
F18 5x5	**39.0**	**36.4**	33.7	31.2	28.5	25.9	23.4
F18, 2 iter.	38.1	35.9	**34.0**	**32.5**	31.3	30.1	29.0
PWMAD 3x3	33.0	32.4	30.7	27.7	24.5	21.6	19.1
PWMAD 5x5	29.0	28.9	28.8	28.4	27.7	26.2	24.0
DWM 5x5	28.8	28.3	27.8	27.2	26.6	25.9	25.0
AMF 5x5	34.3	33.9	33.2	32.2	**31.4**	**30.5**	**29.5**
MF 5x5	26.5	26.4	26.2	26.0	25.8	25.6	25.3
unfiltered	25.1	18.1	15.1	13.3	12.1	11.1	10.3

TABLE III
Comparison of the random valued noise filters in terms of PSNR (dB). The size of the kernel is also specified for each filter.

	noise intensity in percent						
filter	**1**	**5**	**10**	**15**	**20**	**25**	**30**
F17 5x5	**36.0**	**33.4**	30.9	28.7	26.6	24.7	23.0
F17, 2 iter.	34.4	32.5	30.9	**29.6**	**28.4**	**27.2**	25.9
PWMAD 3x3	33.1	32.5	**31.2**	29.5	27.4	25.3	23.3
PWMAD 5x5	29.1	29.0	28.7	28.3	27.8	27.0	**26.0**
DWM 5x5	28.9	28.4	27.8	27.3	26.8	26.2	25.7
AMF 5x5	33.9	30.0	26.0	23.3	21.2	19.6	18.3
MF 5x5	26.6	26.5	26.3	26.1	25.9	25.5	25.2
unfiltered	28.5	21.5	18.5	16.7	15.5	14.5	13.7

A. Salt&Pepper Noise Filters

Table II summarizes the quality of the salt&pepper noise filters. The evolved filter is denoted as F18. As the results show this filter provides the best results for lower noise intensity

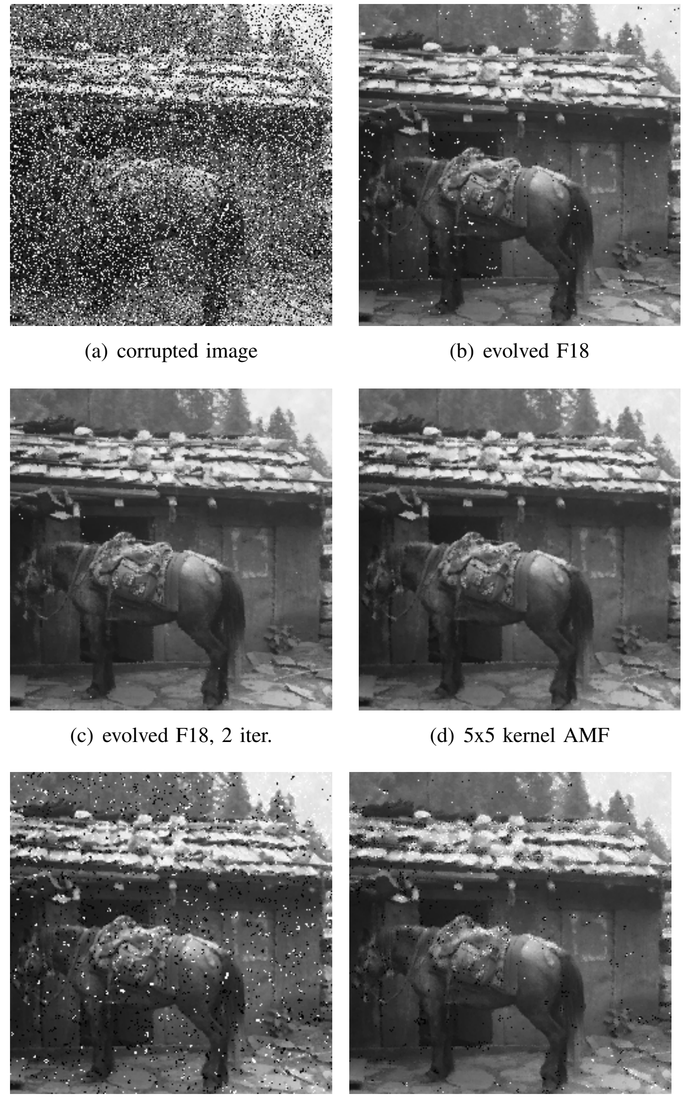


(a) corrupted image (b) evolved F18

(c) evolved F18, 2 iter. (d) 5x5 kernel AMF

(e) 3x3 kernel PWMAD (f) 5x5 kernel PWMAD

Fig. 4. Filtering an image corrupted by 15% salt&pepper noise using different filters



(a) corrupted image (b) evolved F18

(c) evolved F18, 2 iter. (d) 5x5 kernel AMF

(e) 3x3 kernel PWMAD (f) 5x5 kernel PWMAD

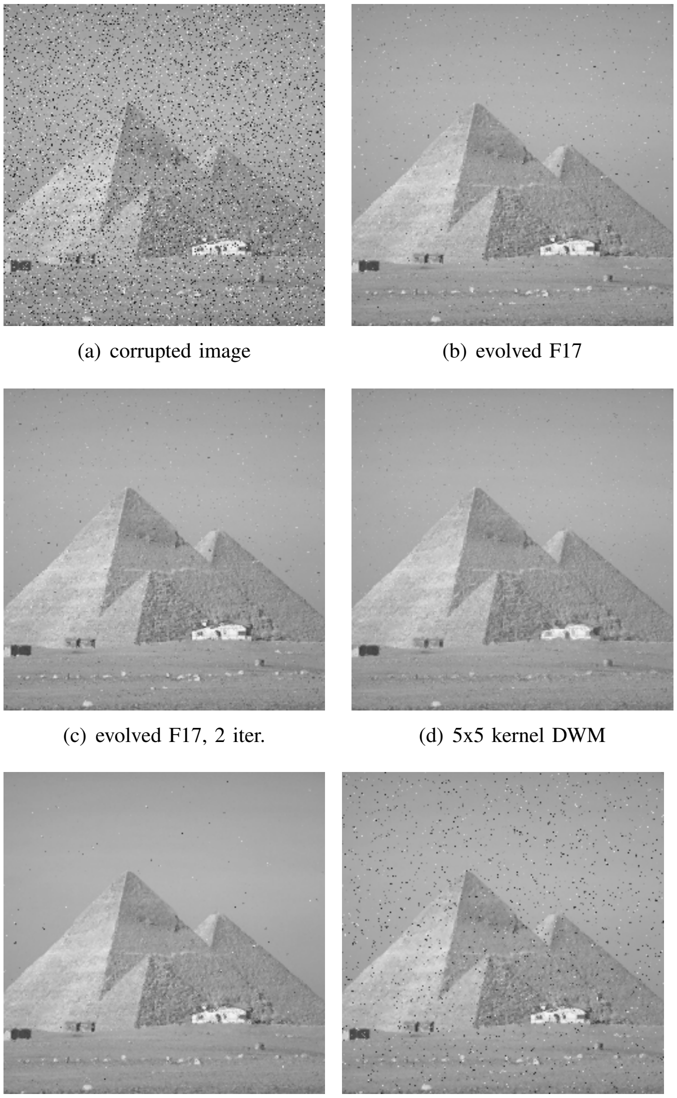
Fig. 5. Filtering an image corrupted by 30% salt&pepper noise using different filters

(1%–15%) in comparison with the conventional filters. For higher noise intensity the AMF results in the highest values of the PSNR (i.e. the best filtering quality from all the compared filters). However, the difference between F18 and AMF for the noise intensity greater or equal to 20% is very small (≤ 0.5).

Figure 4 shows an example of filtering an image corrupted by 15% salt&pepper noise using selected filters. Whilst the evolved filter F18 provides a very good result of the filtered image after a single iteration in case of filtering 15% noise, the iterative PWMAD filter leaves much of the noise in the resulting image (compare the results in Fig. 4b,e,f). The result of the second iteration of F18 (Fig. 4c) is, with respect to the visual quality, very close to the result of 5x5 adaptive median (Fig. 4d).

The comparison of filtering of 30% salt&pepper noise using different filters is shown in Figure 5. Again, the evolved filter F18 provides a high-quality results after the second iteration. The visual quality is comparable to the AMF (see 5c,d). A single application of F18 is not sufficient to obtain the best quality for the image corrupted by 30% noise (as evident in Fig. 5b). However, its results are better in comparison with the PWMAD filter that leaves a lot of noise in the filtered image and makes a loss of some detail (see Figure 5e,f).

In summary, it is possible to say that the CGP-based evolutionary system succeeded in searching a robust salt&pepper noise filter whose filtering quality can compete with the iterative filters and especially the adaptive median filter even for high noise intensity. The main problem of the standard median filter and the DWM filter is that the filtered images are smudged, losing a lot of detail and therefore their outcomes are not included in Figure 4 and 5.

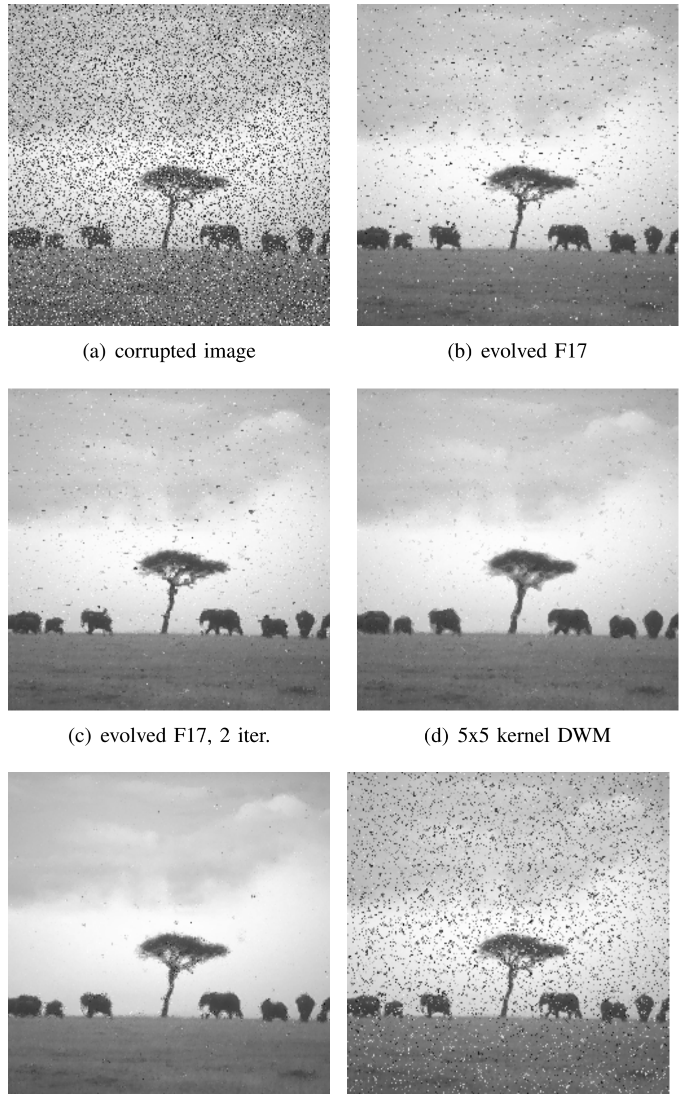


(a) corrupted image (b) evolved F17

(c) evolved F17, 2 iter. (d) 5x5 kernel DWM

(e) 3x3 kernel PWMAD (f) 5x5 kernel AMF

Fig. 6. Filtering an image corrupted by 15% random-valued noise using different filters



(a) corrupted image (b) evolved F17

(c) evolved F17, 2 iter. (d) 5x5 kernel DWM

(e) 5x5 kernel PWMAD (f) 5x5 kernel AMF

Fig. 7. Filtering an image corrupted by 30% random-valued noise using different filters

B. Random-Valued Impulse Noise

The random-valued noise represents a more realistic type of impulse noise in which the corrupted pixels can take an arbitrary value from the entire scale available for the given class of images. Therefore, in this case the noise may be represented by an arbitrary grayscale value from 0 to 255. It is thus more difficult to distinguish the noise pixels among the uncorrupted pixels and the filter may have a higher tendency to deteriorate the overall quality of the filtered image.

Table III shows a comparison of the evolved random-valued noise filter (denoted as F17) with some iterative conventional filters from the point of view of the PSNR. Again, the evolutionary approach succeeded in the search of a robust filter for the noise of this type. The quality of the filtered images using the F17 appears as the best result for all the evaluated filters and most of the noise intensity values.

As expected, the higher noise intensity requires more iterations of the filter to obtain an acceptable result. In case of the F17, at most two iterations were performed. The comparison of the visual quality of different filters considering images corrupted by the random-valued noise is shown in Figure 6 and 7. Very good results can be observed after the second iterations of the evolved filter F17 together with the conventional 5x5 kernel PWMAD. Unlike the case of the salt&pepper noise, the adaptive median filter fails in filtering random-valued noise even for 15% intensity. On the other hand, the conventional DWM filter exhibits a good quality slightly loosing some detail in comparison with the proposed F17 (compare Fig. 7c,d).

C. Comparison with Conventional Iterative Filters

The proposed evolutionary designed filters exhibit some interesting properties in comparison with DWM and PWMAD that were chosen for the comparison. The first aspect represents the iterative filtering process in which the filtered pixel value is not determined in a single step but is calculated in a finite iterative loop. Of course this increases the demands of filtering from the point of view of the computational effort. Whilst DWM and PWMAD required from 5 to 10 iterations (calculated through the entire image), the iterative filtering of the proposed filters computed with just 2 iterations in which even better results were obtained in some cases in comparison with the conventional filters. This demonstrates an ability of the evolutionary process to create an innovative design for the considered typed of noise.

The second point of view of the filters comparison includes the amount of information utilized by the filter to calculate the filtered value. Whilst the conventional filters work with 5x5-pixel kernel (i.e. the filter possesses 25 input values for each pixel to be filtered), all the evolved filters calculate only with a subset of pixels in the filter kernel. The best filter obtained from our experiments possesses only 13 inputs out of the total 25 pixels of the 5x5 kernel. This is a significant difference if compared to the conventional filters which shows that the filtering process may be simple and efficient. One of the advantage of the evolutionary approach is that the target design is based not only on the filter circuit but also on the input selection that allows to optimize the amount of data needed by the filter to work effectively. One of the consequence of this issue is the suitability of the evolved filters for high-speed hardware implementation.

VI. Conclusions

In this paper we proposed an evolutionary algorithm based on the CGP representation to design image filters for impulse noise. The salt&pepper noise and the random-valued impulse noise were considered as case studies. The concept of noise detection (switching filter) was utilized during the evolutionary process to increase the filtering quality and preserve the non-noise pixels to be corrupted by the filtering process. In order to eliminate noise of higher intensity, the resulting filters were applied iteratively and the quality of filtered images were compared to the results obtained by some conventionally used iterative image filters.

In case of the salt&pepper noise, the evolved filters are able to overcome the conventional filters especially for lower noise intensity ($\leq 15\%$). However, the visual quality of the filtered image is very high (after the second iteration of the evolved filter) for higher noise intensity as well and is comparable with the conventional non-iterative adaptive median filter. In this case the iterative filters DWM and PWMAD works even worse in comparison with our filters.

The proposed approach exhibits a very good performance also in filtering the random-valued impulse noise. In this case two iterations of the evolved filter appear sufficient to obtain a result whose quality can compete with the best conventional iterative filters. Although the resulting visual quality of the evolved filter may seem worse, it is able to preserve better the loss of details which is also confirmed by the performed measure in comparison with the conventional solutions (even for higher noise intensity).

We showed that two iterations of the evolved filter is sufficient to obtain a very good result that may overcome the conventional iterative filters that need to perform more iterations to achieve the same quality. One of the crucial issues in the successful evolutionary design is the set of building blocks utilized for the search process. Although the functions used herein for the filter evolution provide very good results, we assume that the design process could be improved by introducing another techniques (in addition to the switching concept or iterative processing) that may require some functions to be changed. Therefore, this idea represents the main objective of our future research.

Acknowledgment

This work was supported by the Czech Science Foundation projects Natural Computing on Unconventional Platforms P103/10/1517 and Mathematical and Engineering Approaches to Developing Reliable and Secure Concurrent and Distributed Computer Systems GD102/09/H042, by the research programme Security-Oriented Research in Information Technology MSM 0021630528 and the BUT grant FIT-S-11-1.

References

[1] E. R. Dougherty and J. T. Astola, Eds., *Nonlinear Filters for Image Processing*, ser. SPIE/IEEE Series on Imaging Science & Engineering. SPIE/IEEE, 1999.

[2] S. Ko and Y. Lee, "Center weighted median filters and their applications to image enhancement," *IEEE Transactions on Circuits and Systems*, vol. 15, pp. 984–993, 1991.

[3] D. R. K. Brownrigg, "The weighted median filter," *Commun. ACM*, vol. 27, no. 8, pp. 807–818, 1984.

[4] S. Marshall, "New direct design method for weighted order statistic filters," *VISP*, vol. 151, no. 1, pp. 1–8, February 2004.

[5] S. T. and Y. Neuvo, "Detail-preserving median based filters in image processing," *Pattern Recognition Letters*, vol. 16, pp. 341–347, 1994.

[6] F. Russo and G. Ramponi, "A fuzzy filter for images corrupted by impulse noise," *IEEE Transactions on Circuits and Systems*, vol. 45, pp. 168–170, 1996.

[7] H. Kog and L. Guan, "A noise-exclusive adaptive filtering framework for removing impulse noise in digital images," *IEEE Signal Processing Letters*, vol. 45, pp. 422–428, 1998.

[8] H. Hwang and R. A. Haddad, "New algorithms for adaptive median filters," in *Proc. SPIE Vol. 1606, p. 400-407, Visual Communications and Image Processing '91: Image Processing, Kou-Hu Tzou; Toshio Koga; Eds.*, K.-H. Tzou and T. Koga, Eds., Nov. 1991, pp. 400–407.

[9] Z. Vasicek and L. Sekanina, "Novel hardware implementation of adaptive median filters," in *Proc. of 2008 IEEE Design and Diagnostics of Electronic Circuits and Systems Workshop*. IEEE Computer Society, 2008, pp. 110–115.

[10] V. Crnojevic, V. Senk, and Z. Trpovski, "Advanced impulse detection based on pixel-wise MAD," vol. 11, no. 7, pp. 589–592, July 2004.

[11] Y. Dong and S. Xu, "A new directional weighted median filter for removal of random-valued impulse noise," *Signal Processing Letters*, vol. 14, no. 3, pp. 193–196, 2007.

[12] R. Lukac, K. Plataniotis, and A. Venetsanopoulos, "Color image denoising using evolutionary computation," *International Journal on Imaging Systems and Technology*, vol. 15, no. 5, pp. 236–251, 2005.

[13] L. Sekanina and V. Drabek, "Automatic design of image operators using evolvable hardware," in *Proc. of 5th IEEE Design and Diagnostics of Electronic Circuits and Systems Workshop*. Brno University of Technology, 2002, pp. 132–139.

[14] J. F. Miller and P. Thomson, "Cartesian Genetic Programming," in *Proc. of the 3rd European Conference on Genetic Programming EuroGP2000*, ser. LNCS, vol. 1802. Springer, 2000, pp. 121–132.