

Evolution of Impulse Bursts Noise Filters

Zdenek Vasicek, Michal Bidlo, Lukas Sekanina

Faculty of Information Technology, Brno University of Technology
Bozotechnova 2, 612 66 Brno, Czech Republic
E-mail: {vasicek,bidlom,sekanina}@fit.vutbr.cz

Jim Torresen, Kyrre Glette, Marcus Furuholmen

Department of Informatics, University of Oslo
P.O. Box 1080 Blindern, N-0316 Oslo, Norway
E-mail: {jimtoer,kyrrehg,marcusfu}@ifi.uio.no

Abstract

The paper deals with evolutionary design of impulse burst noise filters. As proposed filters utilize the filtering window of 5×5 pixels, the design method has to be able to manage 25 eight-bit inputs. The large number of inputs results in an evolutionary algorithm not able to produce reasonably working filters because of the so-called scalability problem of evolutionary circuit design. However, the filters are designed using an extended version of Cartesian Genetic Programming which enables to reduce the number of inputs by selecting the most important of them. Experimental evaluation of the method has shown that evolved filters exhibit better results than conventional solutions based on various median filters.

1 Introduction

Impulse burst noise typically occurs in remote sensing images such as satellite images. The main reason for the occurrence of bursts is the interference of frequency modulated carrying signal with the signals from other data sources. This interference can occur several times during a transmission of a single image and corrupt several image pixels in one or more neighboring rows. Impulse burst noise is often accompanied by salt and pepper noise and multiplicative noise [8].

Various filters have been proposed to suppress this type of noise in the recent years. Some of them were also implemented in hardware. For the purposes of this paper, we can divide these filters into two major groups. The first group will contain general purpose filters for shot noise removal which can be relatively easily implemented in hardware:

median filter [1], adaptive median filter [6] and weight median filter [2]. The second group will consist of specific filters developed for impulse burst noise such as training-based optimized soft morphological filters and variational approaches [9, 8, 13, 3]. Unfortunately, it is much more difficult to implement these filters in hardware than the filters of the first group.

Evolutionary algorithms have been employed to design complete filter structures for various noise types in the recent years. In particular, evolved shot noise filters outperform conventional filters (such as median and adaptive median filters) in terms of the quality of filtering as well as the implementation cost on a chip [15, 16, 17]. Another advantage of filters evolved for the shot noise is that they utilize a small filtering window of 3×3 pixels. The success of the shot noise filters evolution is the main motivation for this paper.

The goal of the paper is to show that efficient filters can be evolved for impulse burst noise. Compared to the evolution of the shot noise filters, we have to deal with several problems. Preliminary experiments have shown that the filtering window 3×3 is too small for this type of noise. Hence proposed filters utilize the filtering window of 5×5 pixels. The increase in the number of inputs from 9 to 25 makes it impossible for the evolutionary algorithm to produce reasonably working filters because of the so-called scalability problem of evolutionary circuit design. In order to overcome this problem, we propose to select only some of the 25 inputs and use them as the inputs to the filtering logic. The filter structure will be designed using Cartesian Genetic Programming (CGP) working at the functional level [14]. The selection of the pixels is controlled by a certain part of the CGP chromosome. The aim is to evolve filters that are suitable for hardware implementation. Furthermore, the method should be portable to a Field Programmable Gate

Array (FPGA) in order to accelerate the design process as presented, for example, in [16, 17].

The rest of the paper is organized as follows. Section 2 surveys the approaches to image filtering, in particular the evolutionary methods used for image filter design. Section 3 describes evolutionary design of image filters using CGP. In Section 4 a modified version of Cartesian Genetic Programming is introduced for evolution of impulse burst noise filters. Section 5 deals with the experimental evaluation of proposed method. Obtained results are discussed in Section 5. Finally, conclusions are given in Section 6.

2 Image filters

Impulse noise in general and impulse burst noise in particular belong to a class of nonlinear noises which have to be suppressed by nonlinear filters.

The median filter is the most popular nonlinear filter for removing the impulse noise [1]. The median filter utilizes the fact that original and corrupted pixels are significantly different and hence the corrupted pixels can easily be identified as non-medians. However, when the noise level increases, some pixels remain corrupted and unfiltered. Although the median filter becomes the most popular approach due to its straightforward hardware implementation, there exist more advanced approaches which exhibit higher filtration quality. The main disadvantage of the median filter is that it modifies almost all pixels and thus produces smudged and detail less image. Among more sophisticated approaches we find switching median filters [20], weighted median filters [2], weighted order statistic filters [10] and adaptive median filters [6].

The adaptive median filters produce significantly better resulting images than conventional medians [7]; however, their implementation cost is relatively high in hardware [19]. The adaptive median filter operates with a kernel of $S_{max} \times S_{max}$ pixels. The kernel is divided into subkernels of the size $3 \times 3, 5 \times 5, \dots, S_{max} \times S_{max}$ inputs. For each subkernel, the minimum, maximum and median value is calculated. In order to obtain the filtered pixel, the calculated values are processed by the algorithm described in [7]. The main difference between the common median filter and the adaptive median filter is that the adaptive median filter tries to detect and modify affected pixels only in order to preserve as much information as possible.

Impulse burst noise is a specific kind of noise which is difficult to filter even if a nonlinear filter is used. This is caused by the fact that both the central pixel and the neighboring pixels are corrupted. It has been shown, the median filters are capable of removing the impulse bursts but at the same time they usually destroy details too heavily. Other filters (e.g. weighted median), are not robust enough and tend to leave a lot of impulse bursts in the filtered images [8].

The approaches based on impulse or spike detection do not perform well in this task since they designed to detect isolated impulses. Apart from median-based filters, training-based optimized soft morphological filters were developed to suppress impulse burst noise [9, 8]. These filters can remove line-type noise with horizontal or almost horizontal orientations.

Besides the conventional algorithms, evolutionary algorithms have been employed in image filter design task. Evolutionary algorithms have been utilized either to find some coefficients of a pre-designed filtering algorithm [4, 11] or to devise a complete structure of a target image filter. As the first approach only allows existing designs to be tuned, the use of the second approach has led to introducing completely new filtering schemes, unknown so far [15]. Evolution of image filters is performed using Cartesian Genetic Programming. A target filter¹ is composed of functional blocks implementing functions such as minimum, maximum, average or elementary logic functions. Each of the blocks accepts two 8-bit operands and produces a single 8-bit output. CGP is utilized to find suitable functions and interconnects for these blocks. The objective of the evolution is to minimize the difference between the image processed by a candidate filter and an uncorrupted version of filtered image. The method was used to produce image filters for various noise (salt and pepper noise, shot noise, Gaussian noise), edge detectors and combinations of noise types. Evolved filters exhibit better quality of filtering and lower implementation cost in comparison to existing solutions (in particular, the median and adaptive median filters) [14, 15, 16, 17, 18].

2.1 Noise model

For the purposes of this paper the impulse burst noise is modeled as follows. The burst noise can be characterized using two parameters: p and q . Let p denote a probability that a certain pixel belongs to an impulse burst. In fact, this parameter determines the maximal amount of the corrupted pixels of an input image. Let q be a parameter which determines the maximal length of burst (i.e. the maximal number of consecutive pixels which are affected by an impulse). The number of burst fragments in the image depends on both these parameters; the higher q , the lower number of burst fragments for a given (constant) value of p .

Figure 1 shows an image (256×256 pixels) which is corrupted by (a) 20% ($p = 0.2$) and (b) 40% ($p = 0.4$) impulse burst noise; the parameter q possesses the value 128. In this paper we will consider that the images are transferred as one-dimensional arrays in which the rows of the image pixels are stored in sequence. The interferences during the image transmission lead to the noise demonstrated

¹Gray-scale images with 8 bit/pixel are considered.

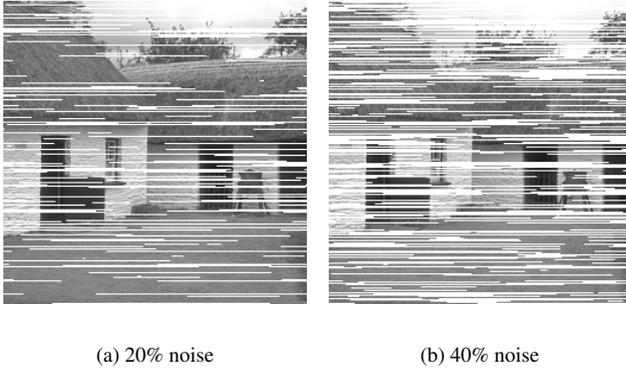


Figure 1. Image (barn) corrupted by impulse burst noise of a various intensity

in Figure 1. The burst noise causes the impulses affecting several pixels in sequence. We consider the impulses which appear as a burst of pixels having the maximal possible value of the given representation (i.e. 255 in the 8-bit grayscale images). With the increased noise intensity, more consecutive rows may be affected and subsequent noise filtration becomes difficult as the filtered value can not be determined according to the values of the neighboring pixels. Therefore, a larger filter window ought to be considered in order to obtain a satisfactory quality of the filtered image. As Figure 1 shows, the burst noise appears as a line-type noise and considerably decreases the image quality especially when higher noise intensity is considered. Depending on the noise intensity, different amount of horizontal strikes appear in those images. In this paper, we assume that images are transferred row-wise, however, the same effect occurs (and the same filtering approach can be applied) if images are transferred column-wise.

3 Evolutionary Design of Image Filters

This section describes the evolutionary method that has been utilized to create image filters with 3×3 pixel filter window [15]. In Section 4, we will describe an enhanced approach by means of which the filters dealing with larger filter window could be designed. Every image filter is considered as a function (a digital circuit in the case of hardware implementation) of nine 8-bit inputs and a single 8-bit output, which processes grayscale (8-bit/pixel) images. As Fig. 2 shows, every pixel value of the filtered image is calculated using a corresponding pixel and its eight neighbors in the processed image.

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) an original (training) image to measure the fitness

values of the candidate filters (i.e., to evaluate the quality of a candidate filter). 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.

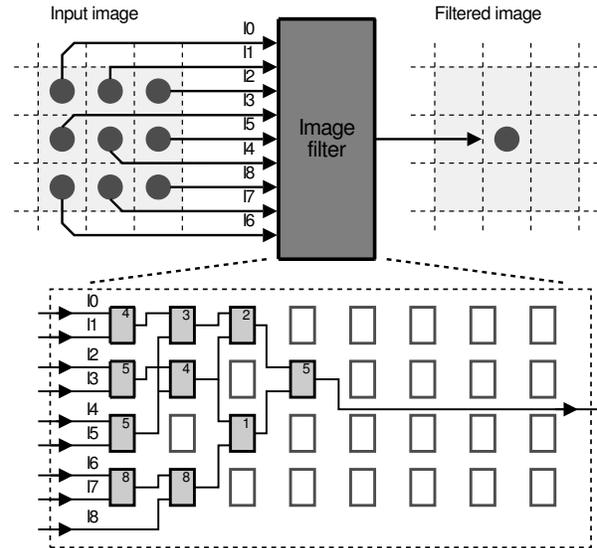


Figure 2. The concept of filtering system that utilizes 3×3 filter window

3.1 EA for filter evolution

The evolutionary design of the filter circuits is based on Cartesian Genetic Programming (CGP) [12]. A candidate filter is represented using a graph which contains n_c (columns) \times n_r (rows) nodes placed in a grid. The role of the EA is to find the interconnection of the programmable nodes and the functions performed by the nodes. Each node represents a two-input function that receives two 8-bit values and produces an 8-bit output. Table 1 shows the functions we identified as suitable building blocks for the evolutionary design of the impulse burst noise filters. It can be observed that the functions are convenient for hardware implementation (there are no functions that are very time-consuming and require a lot of hardware resources, e.g. multiplication or division). A node input may be connected either to an output of another node, which is placed anywhere in the preceding columns or to a primary input of the filter. The filter circuits are encoded as arrays of integers of the size $3 \times n_r \times n_r + 1$. For each node, three integers are utilized which encode the connection of the node inputs and function. The last integer of the array encodes the primary output of the candidate filter.

Table 1. The list of functions that can be implemented in each configurable 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	$x \gg 1$	right shift by 1
6	$x \gg 2$	right shift by 2
7	$x + y$	+ (addition)
8	$x +^S y$	+ with saturation
9	$(x + y) \gg 1$	average
10	y if $(x > 127)$ else x	condition
11	$ x - y $	absolute difference
12	$x \vee y$	bitwise OR
13	$x \wedge y$	bitwise AND
14	$x \oplus y$	bitwise XOR
15	$\overline{x \wedge y}$	bitwise NAND

3.2 Fitness function

The design objective is to minimize the difference between the filtered image and the original image. Usually, the *mean difference per pixel* also known as the *mean absolute error* is minimized. Let u denote a corrupted image, v the filtered image and w the original (uncorrupted) version of u . The image size is $K \times K$ ($K=256$) pixels but only the area of 254×254 pixels is considered because the pixel values at the borders are ignored and thus remain unfiltered. The fitness value of a candidate filter is obtained by calculating the error function:

$$fitness = \frac{1}{(K-2)^2} \sum_{i=1}^{K-2} \sum_{j=1}^{K-2} |v(i, j) - w(i, j)|.$$

The objective is to design a filter producing images with minimal error, i.e. the lower fitness value the better filter. Note that practically it is impossible to obtain a filter possessing the zero fitness value (i.e. an ideal filter) since the filter manipulates with corrupted images only (i.e. missing and incomplete information) and it can not predict the original values perfectly for an arbitrary input image. Only in rare cases (e.g. a training image with simple pattern), it is possible to evolve filter that possesses the zero fitness value but this filter will not be probably robust (i.e., it will work only for the selected training image). However, if a suboptimal solution fulfills a given criterion of quality, it is usually considered as a solution to the problem.

It is evident that the robustness of evolved filter depends on the selection of the training data. In our previous research, it has been determined that an image containing 128×128 pixels provides the sufficient amount of training data for evolution of robust 3×3 filters. As we utilize a

larger filter window in this work, we have chosen the training image consisting of 256×256 pixels.

4 Evolving burst noise filters

In order to create a robust burst noise filter, it is necessary to increase the size of the filter window. Since the burst noise affects a series of pixels (rather than only isolated pixels), more information from the pixels surrounding the corrupted pixels is needed to remove this type of noise and obtain a satisfactory quality of filtered images. However, if the size of the filter window is increased (in our case from 3×3 to 5×5), the search space that is needed to be explored using EA grows enormously as each input of the programmable node can connect one of the 25 primary inputs. Preliminary experiments showed that the evolved filter rarely utilizes more than a half of the number of pixels inside the filter window. In order to overcome the scalability problem of the utilized representation which employs the 5×5 filter window, we propose the following approach.

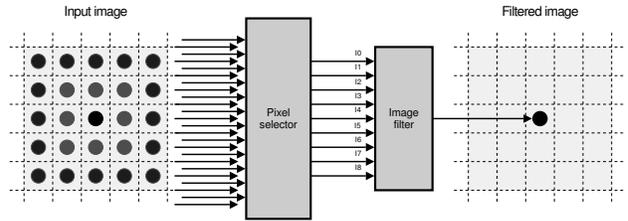


Figure 3. The concept of filtering using a 5×5 filter kernel followed by a selector

A special pixel selection mechanism (a selector) is introduced into the filter to be evolved that allows us to choose a subset of pixels from a larger filter window. Another architecture employing the concept of selecting data elements before passing them to a CGP circuit, in this case for a pattern recognition system, has been proposed in [5]. The concept of selector is illustrated in Figure 3. The output size of the selector is specified by a parameter that determines the number of pixels to be selected from the filter window (i.e. processed by the filter). Let us denote this parameter as S . This method reduces the search space and computational effort needed to design a filter while keeping the possibility to provide information to the filter circuit from anywhere of a given portion inside the filter window.

For example, consider the 5×5 filter window consisting of 25 pixels from which only 9 pixels need to be included in the filtering process (i.e. $S = 9$). This approach clearly reduces the search space because each functional node may select only from 9 inputs instead of 25 inputs of the filter window.

4.1 Designing filters with selector

In order to integrate the selector into the design system, it is needed to encode it as a part of the chromosome of the evolutionary algorithm. Then the EA has to find (1) the circuit realizing the filter and (2) a suitable variation of indexes from a given subset of the pixels in the filter window. The following scheme will be utilized.

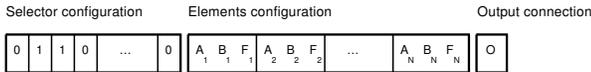


Figure 4. Proposed encoding

Each chromosome of the EA encodes the selector together with the string describing a circuit using CGP. Figure 4 demonstrates the proposed encoding. The chromosome consists of three parts. In the first part, the selector mechanism is encoded, the second part contains the CGP representation of the evolving filter (i.e. configuration of each node and their interconnections) and the last third part encodes the connection of the output. The selector is encoded as a binary string that contains the same number of bits as the number of pixels in the filter window (i.e. the selector string consists of 25 bits for the 5x5 filter window). In order to distinguish the pixels to be selected, the values of the appropriate bits are set to logic 1 while the rest of the bits possess logic 0. Each bit of the selector string (considered from left to right) gradually corresponds to the appropriate pixel of the filter window (considered from left to right and from top to bottom). The number of logic 1's corresponds to the selector parameter S . The pixel selection process is performed as follows. If the bit of the selector string corresponding to the given pixel of the filter window possesses logic 1, then the pixel will be selected, otherwise the pixel will not be considered as an input of the filter.

A special mutation operator is utilized for the effective modification of the selector. This operator takes as an input two different indexes from the range 0 to $S - 1$ according to which the appropriate bits of the selector string are swapped. If a bit possessing logic 0 is swapped with a bit possessing logic 1, then the selector is altered and a new combination of pixels will be selected from the filter window on the basis of the 1's arrangement in the selector string. Otherwise the mutation represents, in fact, a neutral genetic operation that does not influence the candidate filter.

4.2 Evolutionary system setup

The evolutionary algorithm utilizes a single genetic operation – the mutation – which may modify up to 5% of genes of the chromosome (this value was determined experimentally). The index of a gene to be mutated is generated at

random. If the index hits one of the first 25 genes representing the selector string, a second index is generated and the mutation of the selector is performed as described in the previous paragraph. Otherwise, if the elements configuration or output connection encoded in the rest of the chromosome should be mutated, the value of the given gene is replaced by a new random legal value. No crossover operator is utilized in this type of EA. The EA operates with the population of 8 individuals. The initial population is generated randomly. Every new population consists of a parent, which is considered as the fittest individual from the previous population, and its mutants. In the case that two or more individuals have received the same fitness score in the previous generation, an individual which did not serve as the parent in the previous population will be selected as the new parent. This strategy was proven to be very useful for the evolution using the CGP [12]. The single run of the evolutionary process is stopped after producing 50,000 generations. We performed 150 independent experiments for each setup (i.e., for the 3×3 and 5×5 filter window).

A candidate filter can use up to 9 pixels selected from the 5×5 filter window. The filter may consist of up to 6×6 programmable nodes, each of which can possess the functions from Table 1. A 256×256 -pixel training image was corrupted by 20% burst noise. This image is shown in Figure 1a.

5 Experimental results

The experiments were conducted on a cluster consisting of 100 PCs Pentium IV, 2.4GHz, 1GB RAM using the Sun Grid Engine (SGE) that allows to run up to 100 independent experiments in parallel. The evolution time of a single run until the evolution reaches 50,000 generations is approximately 6 hours.

5.1 Evaluation of evolved filters

The evolved filters are evaluated using a set of real-world images (256×256 pixels) that are corrupted by aforementioned impulse bursts noise with the intensity of 10%, 20%, 30%, 40% and 50%. This corresponds to the following parameters of the noise model: $p = \{0.1, 0.2, 0.3, 0.4, 0.5\}$, the parameter w has been chosen as $w = 128$. The test set consists of the images having various properties – some of them contain a lot of details and complex structures (e.g. goldhill, bridge) some of them are easier to filter as they do not contain many edges and details.

In order to evaluate the quality of the filtered images, we will use the following widely used metrics: the mean absolute error (MAE) and the peak signal to noise ratio (PSNR). Let w denote the original 8-bit gray scale image, v denote the filtered image (after suppressing the noise) and $M \times N$

Table 2. The MAEs and PSNRs between original test images and corrupted images filtered by various methods.

image	filter	MAE					PSNR				
		noise intensity					noise intensity				
		10%	20%	30%	40%	50%	10%	20%	30%	40%	50%
airplane	median 5 × 5	7.28	12.67	20.01	32.83	44.04	23.63	18.75	16.08	13.06	11.84
	weighted 5 × 5	5.38	11.06	18.65	32.04	43.55	24.83	19.03	16.22	13.09	11.86
	adaptive 5 × 5	2.29	6.36	12.02	24.02	34.27	27.99	20.98	18.03	14.25	12.85
	evolved 3 × 3	4.29	5.42	7.16	10.44	13.96	24.90	23.76	22.40	20.49	19.10
	evolved 5 × 5	0.87	1.77	2.81	4.55	7.25	33.51	29.96	27.38	23.91	21.01
	original error	7.10	15.46	22.53	31.19	38.64	19.69	15.98	14.57	12.89	12.12
bird	median 5 × 5	3.98	10.10	25.33	45.36	67.81	25.42	17.93	13.19	10.18	8.32
	weighted 5 × 5	3.10	9.25	24.78	44.82	67.38	25.66	18.05	13.17	10.19	8.33
	adaptive 5 × 5	1.75	6.30	18.51	34.12	53.89	27.61	19.38	14.33	11.33	9.25
	evolved 3 × 3	0.59	1.18	2.48	4.12	6.17	36.14	32.77	28.67	25.67	23.73
	evolved 5 × 5	0.26	0.55	1.18	2.65	5.38	44.76	40.80	30.68	24.58	20.41
	original error	13.05	26.23	38.14	52.11	65.31	15.33	12.29	10.78	9.31	8.32
bridge	median 5 × 5	13.38	20.69	35.98	57.78	80.20	21.17	16.05	12.09	9.14	7.42
	weighted 5 × 5	10.27	18.12	34.26	56.31	79.40	22.08	16.18	12.06	9.15	7.41
	adaptive 5 × 5	4.59	10.58	22.96	41.54	62.50	24.94	17.85	13.58	10.33	8.37
	evolved 3 × 3	1.84	3.41	5.64	8.30	12.22	29.44	26.55	24.13	22.21	20.12
	evolved 5 × 5	1.83	3.39	5.29	7.99	12.55	29.60	26.60	24.15	21.22	17.79
	original error	14.38	28.38	42.37	56.34	71.01	14.46	11.54	9.83	8.56	7.55
goldhill	median 5 × 5	9.83	18.93	35.31	54.96	79.20	22.94	15.75	11.61	9.18	7.41
	weighted 5 × 5	7.50	17.08	33.81	53.76	78.41	23.78	15.84	11.63	9.20	7.41
	adaptive 5 × 5	3.48	11.31	24.00	40.28	63.21	26.93	17.01	12.89	10.31	8.30
	evolved 3 × 3	1.47	2.55	4.61	6.34	9.41	30.25	28.66	25.30	23.78	21.75
	evolved 5 × 5	0.90	2.03	3.17	5.87	8.97	35.62	30.68	28.10	22.15	19.50
	original error	14.23	28.28	43.04	58.13	71.85	14.61	11.59	9.75	8.41	7.53
lena	median 5 × 5	7.68	15.73	31.42	52.04	77.72	22.69	16.49	11.89	9.32	7.25
	weighted 5 × 5	5.70	14.09	30.09	50.96	76.71	23.22	16.54	11.90	9.31	7.27
	adaptive 5 × 5	2.77	8.98	20.86	36.88	62.50	25.89	18.05	13.28	10.61	8.09
	evolved 3 × 3	1.47	2.47	4.15	6.26	10.72	28.17	26.79	24.31	22.58	19.83
	evolved 5 × 5	0.86	1.58	2.66	3.94	8.94	32.76	30.70	27.03	24.24	18.12
	original error	14.64	28.47	43.63	57.71	72.63	14.23	11.42	9.50	8.32	7.30

express the size of the image. Then the mean absolute error can be expressed as

$$MAE = \frac{1}{MN} \sum_{i,j} |w(i, j) - v(i, j)|$$

and the peak signal to noise ratio can be expressed as

$$PSNR = 10 \log_{10} \frac{255^2}{\frac{1}{MN} \sum_{i,j} (w(i, j) - v(i, j))^2}$$

In general, the lower value of MAE, the better quality of the filtered image. On the contrary, the higher value of PSNR, the better quality of the filtered image.

Table 2 shows the resulting values (i.e. MAE and PSNR) for the five images selected from the test set filtered using several approaches. The resulting images obtained by the evolved filters are compared to the results of three conventional approaches that include traditional median filter (denoted as ‘median’ in Table 2), center weighted median filter considering the weight 5 (denoted as ‘weighted’) and adaptive median filter (denoted as ‘adaptive’). Since these filtering methods provide low-quality of the filtered images if

the 3 × 3 filter window is used, only 5 × 5 window will be included for the comparison with the evolved filters.

These results provided by the filters designed using an evolutionary algorithm are described as ‘evolved’ in Table 2. In the first set of experiments, 3 × 3 filter window was considered (without the selector) and the second set of experiments utilized 5 × 5 filter window. The results include the best filters that were obtained using our evolutionary approach.

In general, the center weighted median filter provides sharper images in comparison with traditional median filter. The adaptive median represents the best median-based filter with respect to the quality of the filtered images.

As the results from Table 2 show, the evolved filters represent the best solutions in comparison with the conventional principles (the best result for the given noise intensity and the criterion applied is written in bold). If the 5 × 5 filter window is utilized, the filters exhibit better results against the 3 × 3 filter window in most cases with respect to the evaluation criteria applied. For the comparison of the filtration quality using different filters, the original error is

included in Table 2 which represents the errors calculated using the corrupted image and the original image. It is evident that the filters utilizing the 5×5 window provide the best results in comparison with the 3×3 filter window from the point of view of the MAE. If the PSNR is considered, the filter utilizing the 3×3 window results as the best solution in several cases for high noise ratio but not from the point of view of the visual quality of the filtered images (see discussion in the next subsection).

5.2 Discussion

Although the evolved filters represent the best solutions in comparison with some widely used median-based filters, the proposed approach does not guarantee that an optimal solution will be obtained. However, as mentioned in Section 3.2, the suboptimal solutions are usually satisfactory considering the visual quality of the filtered images.

If the 3×3 filter window is considered, the best evolved filters were not able to remove the impulse bursts sufficiently (see Figure 5c). In particular, these filters are not able to tackle several bursts occurring in consecutive rows because of the small filter window. In such cases, most of those bursts remain in the filtered image. It can be observed that the evolution has designed solutions in which the corrupted pixels of the impulse bursts possessing the value 255 (a white) are replaced by the value 128 (gray) which leads to better value of PSNR. This feature can be observed in Table 2, where the filter utilizing the 3×3 window exhibits better PSNR in comparison with the filter that involves the 5×5 window, especially for higher values of the noise intensity. However, this property does not lead to better visual quality of the filtered images because most of the bursts whose pixels were substituted by the value 128 remain in the resulting image (compare the filtered images in Figure 5c and 5d). Therefore, it is more suitable to consider MAE instead of PSNR as the evaluation criterion.

As expected, the filters that utilize the 5×5 window are better in general in comparison with the 3×3 window filters if the visual quality of the filtered images is considered.

As Figure 6 shows, the evolution typically designs the selector that considers only such pixels from the filter window that possess the sufficient amount of information needed for the filter to remove the impulse bursts (the middle column of the filter window). As evident, the bursts do not provide any information in the horizontal direction for calculating the correct values of the filtered pixels. Therefore, such pixels from the filter window have not been considered by the filter. Note that the evolved filter shown in Figure 6 was optimized by removing the meaningless functions (e.g. by omitting the identity functions and removing all the blocks whose outputs are not involved in the calculation of the value of the filtered pixel).

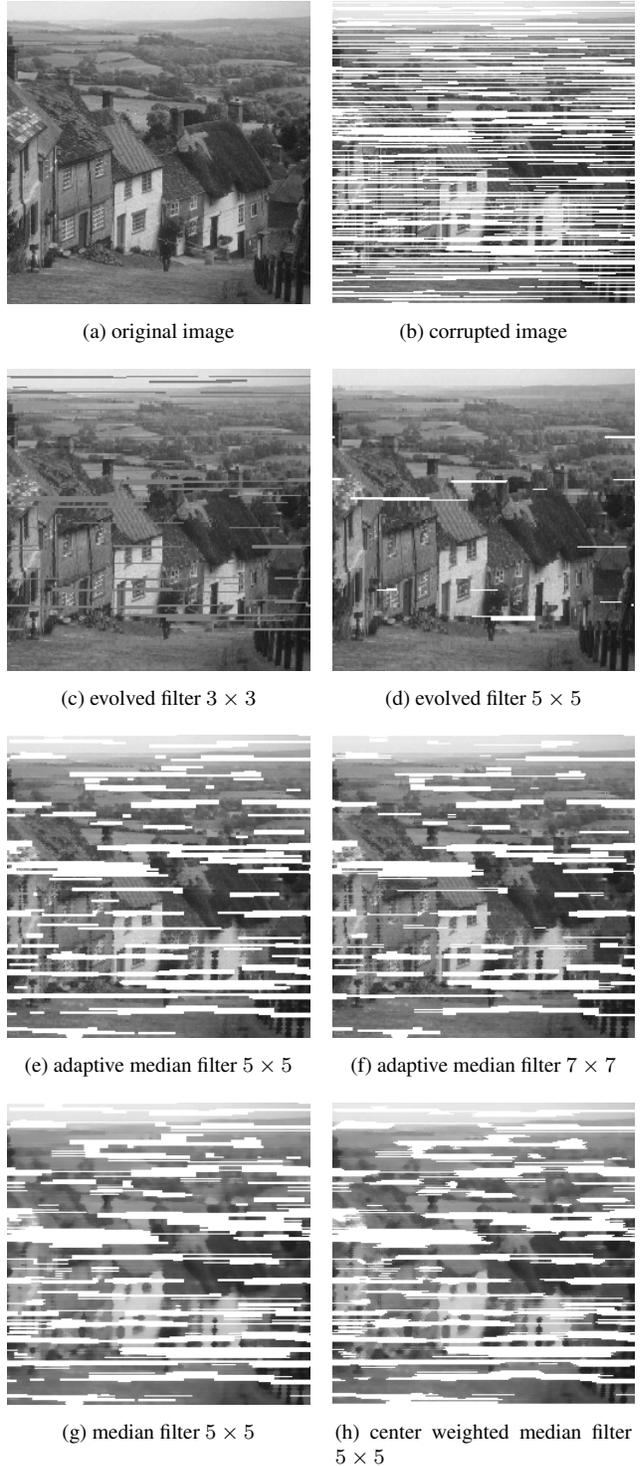


Figure 5. Image (goldhill) corrupted by 40% impulse burst noise filtered by various approaches

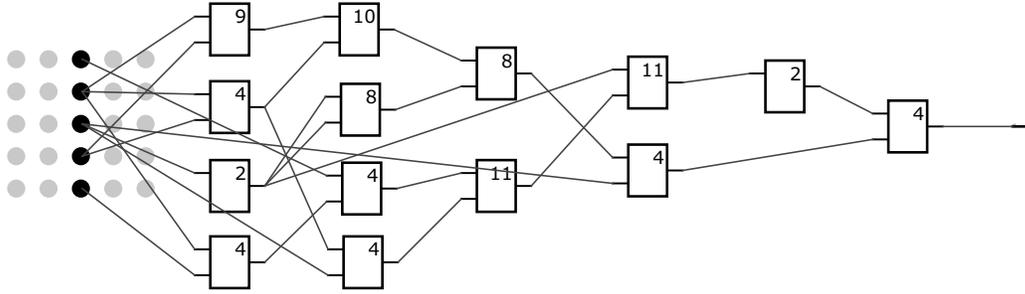


Figure 6. Optimized structure of the best evolved 5×5 filter

6 Conclusions

In this paper, a new method for design of impulse bursts noise filters was proposed. The solution is based on extended version of CGP that can effectively deal with large filtering windows that have to be utilized in this task. Experimental evaluation of the method has shown that evolved filters exhibit better results than conventional solutions based on various median filters. In our future work we plan to implement some of the evolved filters in the FPGA since the implementation requirements seem to be reasonable for the hardware implementation.

Acknowledgment

This work was partially supported by the Grant Agency of the Czech Republic under contract No. GA102/07/0850 *Design and hardware implementation of a patent-invention machine*, No. GD102/09/H042 and the Research Plan No. MSM0021630528

References

- [1] M. O. Ahmad and D. Sundararajan. A fast algorithm for two-dimensional median filtering. *IEEE Transactions on Circuits and Systems*, 34:1364–1374, 1987.
- [2] D. R. K. Brownrigg. The weighted median filter. *Commun. ACM*, 27(8):807–818, 1984.
- [3] E. R. Dougherty and J. T. Astola, editors. *Nonlinear Filters for Image Processing*. SPIE/IEEE Series on Imaging Science & Engineering. SPIE/IEEE, 1999.
- [4] J. Dumoulin, J. Foster, J. Frenzel, and S. McGrew. Special Purpose Image Convolution with Evolvable Hardware. In *Proc. of the EvoIASP'00*, volume 1803 of *LNCS*, pages 1–11. Springer Verlag, 2000.
- [5] K. Glette, J. Torresen, M. Yasunaga, and Y. Yamaguchi. On-Chip Evolution Using a Soft Processor Core Applied to Image Recognition. In *Proceedings 1st NASA /ESA Conference on Adaptive Hardware and Systems (AHS)*, pages 373–380. IEEE CS Press, 2006.
- [6] H. Hwang and R. Haddad. Adaptive median filters: new algorithms and results. *IP*, 4(4):499–502, April 1995.
- [7] H. Hwang and R. A. Haddad. New algorithms for adaptive median filters. In *Proc. SPIE Vol. 1606*, pages 400–407, 1991.
- [8] P. Koivisto, J. Astola, V. Lukin, V. Melnik, and O. Tsymbal. Removing Impulse Bursts from Images by Training-Based Filtering. *EURASIP Journal on Applied Signal Processing*, 2003(3):223–237, 2003.
- [9] P. Koivisto, H. Huttunen, and P. Kuosmanen. Training-based optimization of soft morphological filters. *Journal of Electronic Imaging*, 5(3):300–322, 1996.
- [10] S. Marshall. New direct design method for weighted order statistic filters. *VISP*, 151(1):1–8, February 2004.
- [11] S. Marshall, N. Harvey, and D. Greenhalgh. Design of morphological filters using genetic algorithms. In *Tenth European Signal Processing Conference*, pages 389–392. EURASIP, 2000.
- [12] J. F. Miller and D. Job. Principles in the evolutionary design of digital circuits – part I. *Genetic Programming and Evolvable Machines*, 1(1):8–35, April 2000.
- [13] M. Nikolova. A variational approach to remove outliers and impulse noise. *J. Math. Imaging Vis.*, 20(1-2):99–120, 2004.
- [14] L. Sekanina. Image filter design with evolvable hardware. In *Applications of Evolutionary Computing*, volume 2002, pages 255–266. Springer Verlag, 2002.
- [15] L. Sekanina. *Evolvable components: From Theory to Hardware Implementations*. Natural Computing. Springer-Verlag Berlin, 2004.
- [16] Z. Vasicek and L. Sekanina. An area-efficient alternative to adaptive median filtering in fpgas. In *Proc. of the 17th Conf. on Field Programmable Logic and Applications*, pages 1–6. IEEE Computer Society, to appear, 2007.
- [17] Z. Vasicek and L. Sekanina. An evolvable hardware system in xilinx virtex ii pro fpga. *International Journal of Innovative Computing and Applications*, 1(1):63–73, 2007.
- [18] Z. Vasicek and L. Sekanina. Reducing the area on a chip using a bank of evolved filters. In *Evolvable Systems: From Biology to Hardware*, volume 4684 of *LNCS*, pages 222–232. Springer Verlag, 2007.
- [19] 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*, pages 110–115. IEEE Computer Society, 2008.
- [20] W. Zhou and Z. David. Progressive switching median filter for the removal of impulse noise from highly corrupted images. *IEEE Trans On Circuits and Systems: Analog and Digital Signal Processing*, 46(1):78–80, 1999.