Medical image denoising on field programmable gate array using finite Radon transform

A. Ahmad, A. Amira, H. Rabah, Y. Berviller

Medical image denoising on field programmable gate array using finite Radon transform

Article *in* Signal Processing, IET · December 2012

DOI: 10.1049/iet-spr.2011.0392

CITATIONS
6

READS
169

4 authors, including:

Afandi Ahmad
Universiti Tun Hussein Onn Malaysia
48 PUBLICATIONS **277** CITATIONS
SEE PROFILE

Abbes Amira
University of Sharjah
393 PUBLICATIONS **5,243** CITATIONS
SEE PROFILE

Yves Berviller
University of Lorraine
53 PUBLICATIONS **224** CITATIONS
SEE PROFILE

www.ietdl.org

Published in IET Signal Processing
Received on 25th October 2011
Revised on 17th August 2012
doi: 10.1049/iet-spr.2011.0392

ISSN 1751-9675

Medical image denoising on field programmable gate array using finite Radon transform

A. Ahmad[1] *A. Amira*[2,3] *H. Rabah*[4] *Y. Berviller*[4]

[1]*Department of Computer Engineering, Facutly of Electrical & Electronic Engineering, Universiti Tun Hussein Onn Malaysia (UTHM), Batu Pahat, Malaysia*
[2]*NIBEC, University of Ulster, Jordanstown Campus, Newtownabbey, United Kingdom*
[3]*Department of Electrical Engineering, Qatar University, Doha, Qatar*
[4]*Department N2EV, Institut Jean Lamour, University of Lorraine, Nancy, France*
E-mail: a.amira@ulster.ac.uk

Abstract: This study presents the design and implementation of efficient architectures for finite Radon transform (FRAT) on a field programmable gate array (FPGA). FPGA-based architectures with two design strategies have been proposed: direct implementation of pseudo-code with a sequential or pipelined description, and a block random access memory-based approach. Various medical images modalities have been deployed for both software evaluation and hardware implementation. Xilinx DSP tool has been used to improve the implementation time and reduce the design cycle and the Xilinx software has been used for generating a hardware description language from a high-level MATLAB description. Objective evaluation of image denoising using FRAT is carried out and demonstrates promising results. Moreover, the impact of different block sizes on image reconstruction has been analysed. Performance analysis in terms of area, maximum frequency and throughput is presented and reveals significant achievements.

1 Introduction

In medical imaging systems, noise can be classified as additive or multiplicative [1]. Noise reduction in medical imaging applications is very important, as various types of noises generated by medical imaging equipment, consequently, limit the effectiveness of medical image diagnosis [2].

The contributions of transform domains in various applications including image denoising, enhancement and compression are undebatable facts. As an example, the wavelet transform has been extensively used as a solution to the problem of the short time Fourier transform (STFT) and excels in isolation discontinuities and spikes [3]. However, the wavelet suffers from flexible directionality, as it does not isolate the smoothness along edges. This demerit of the wavelet is well addressed by the ridgelet and curvelet transforms, as they extend the functionality of the wavelets to higher dimensional singularities, and it is proven as an effective tool to perform sparse directional analysis [3]. The basic building block of these transforms is the finite Radon transform (FRAT).

Since medical images contain several objects and curves, doubtless, the curvelet and ridgelet with their main building block FRAT play a major role in better image analysis. By offloading the intensive processing procedures of these transforms into a proper hardware platform, computational acceleration can be achieved, and at the same time the outcomes quality can be maintained.

The FRAT algorithm is stringent, as it is inherently serial, iterative and has a long latency. To overcome these boundaries, there is a real need for hardware implementation and acceleration of FRAT especially for medical imaging applications. Existing limited hardware implementation of the FRAT in medical imaging applications opens a huge gap to be filled [3–7].

This paper presents the design and implementation of FRAT on reconfigurable hardware using a field programmable gate array (FPGA) for medical image denoising using the Xilinx DSP tool. Two design strategies have been proposed: direct implementation of pseudo-code with a sequential or pipelined description and block random access memory (BRAM)-based method. Analysis for both software simulation and hardware implementation with different medical image modalities has been carried out and discussed. An evaluation of FRAT's capability on medical image denoising is also addressed.

The organisation of this paper is as follows. The related work is presented in Section 2. An overview of the algorithms used is presented in Section 3. Section 4 explains the proposed system implementation in two aspects: denoising system and architectures. Experimental results analysis of medical image denoising, using software simulations and hardware implementations are explained in Section 5. Finally, a summary is given in Section 6.

1.1 Related work

Several existing hardware implementations of FRAT are discussed in this section. In [4], two architectures are proposed, a generic and a standard FRAT-based pseudo-code.

The generic architecture uses a combination of look-up tables, matrix of accumulators and multiplexers to perform the FRAT with time complexity $O(p^4)$, where p is the block size in pixels. The second FRAT architecture is based on the standard FRAT pseudo-code presented in [8] with a time complexity of $O\ (p^4\ (p+1))$.

Two architectures for FRAT suitable for FPGA implementation are described in [5]. The first architecture is a direct hardware implementation of a suitable modified variant of the standard FRAT pseudo-code [8], called the reference FRAT architecture. This reference architecture comprises of an address logic initialiser (ALI), multiplexer, accumulators and two memory blocks for storing transform vectors. The second architecture is denoted as a memoryless FRAT architecture, which operates in a parallel manner with p times the throughput of the first architecture. The proposed architectures use 7×7 size image blocks and are prototyped for processing a common intermediate format image sequence. The simulation and synthesis results on Xilinx Virtex-II FPGA show that the core speeds of the two proposed architectures are around 100 and 82 MHz, respectively.

In [6], the proposed FRAT architecture is a direct hardware implementation of the pseudo-code that uses ALIs and a controller to control the accumulators and the memory blocks. The ALI along with controller block constitutes the address generator that generates addresses, that is, $L_{k,l}$ for memory blocks. The accumulator is an L_o bit accumulator that accumulates the Lth pixel value for the Kth Radon projection. The design has been ported to the Xilinx Virtex-II FPGA chip using Handel-C. Moreover, there was no specific optimisation performed on the Radon block, since this work is mainly for finite rigdelet transform (FRIT) implementation.

In [7], two power efficient architectures for the FRAT with time complexities $O\ (p^2\ (p+1))$ and $O\ (p^2)$, respectively, are proposed. The first architecture is a serial architecture based on the FRAT pseudo-code and optimised for FPGA implementation. The second architecture is based on a parallelised version of the FRAT pseudo-code and consists of a combination of read only memory-based control logic and array-based buffers for input/outputs (I/Os).

The merit of a serial input architecture is that the FRAT block can be easily included into a sequence of image processing/compression steps such as the ridgelet or curvelet, without imposing any restrictions on the nature of the inputs. No clock cycles are wasted in buffering the whole input block and the input section can be pipelined. For the second architecture, the input buffer is a linear distributed RAM with p^2 address locations, with p as the block size. On the contrary, the output buffer is a linear array of shift registers with p locations. Both architectures are implemented on the Virtex-E FPGA series and prototyped on the Celoxica RC1000 development board.

The most recent implementation of the FRAT is described in [3]. A parametrisable, scalable and high-performance core of the FRAT sub-block was presented. To provide with design reusability, the core of the FRAT is developed using Handel-C and an efficient VHDL core from Xilinx CoreGen. Two main strategies are imposed in this architecture: re-mapping the FRAT pseudo-code in the order of input signals and implementing a systolic array to store the address de-referencing values rather than using multiplexer or counter chains. The design is prototyped on the Celoxica RC1000 board containing the Xilinx XCV2000E FPGA and the results obtained reveal the design time and area complexity as $O\ (p^2)$ and $O\ (2p^2)$, respectively.

Table 1 Summary of FFPGA-based architectures of FRAT

Refs.	FPGA devices	Programming approaches	Target applications
[3]	Virtex-E	Handel-C and CoreGen	image processing
[4]	Virtex-E	N/A	image processing
[5]	Virtex-II	Verilog	image processing
[6]	Virtex-II	Handel-C	image processing
[7]	Virtex-E	Handel-C	image processing

Table 1 lists the important issues, such as the FPGA devices used along with the programming approaches as well as the target applications for different existing systems. In terms of programming approaches, Handel-C seems to be the most popular tool that has been used, because the target FPGA device changes in line with FPGA technology advancement. For the target applications, previous works reported are mainly for general image processing evidenced by the type of images used. To clearly show innovation and improvement from the existing implementation of FRAT on FPGA, each contribution with its important features and invention is depicted in the research timeline as shown in Fig. 1.

Hardware implementation of image denoising: Software simulation of image denoising [1, 2, 9–13] has gained much effort from the research community when compared with hardware implementation.

In [14], the parallel implementation of an advanced wavelet-domain noise filtering algorithm, which uses a non-decimated wavelet transform and spatially adaptive Bayesian wavelet shrinkage, is discussed. The system implemented is computationally intensive; hence two FPGAs have been deployed to work in parallel. Partial reconfiguration could have been taken into account for better system implementation as well as maintaining the same performance.

An FPGA implementation of a block singular value decomposition (SVD) method for image denoising is presented in [15]. This method exploits the fact that only the smallest SVD eigenvalues are affected by the noise and therefore can be discarded leading to an efficient non-linear image filtering. The architecture is based on the Brent Luk Van (BLV) loan systolic array technique.

Another implementation of a flexible hardware architecture for performing the DWT on a digital image for denoising is addressed in [16]. A simple wavelet technique demonstrates some improvement with a peak signal to noise ratio (PSNR) up to 27 dB. The DWT core is modelled using MATLAB and VHDL and implemented on Xilinx FPGA devices. Results for hardware implementation show that about

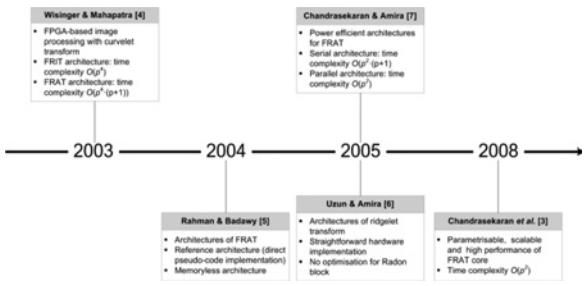

Fig. 1 *Review of FRAT's FPGA-based implementation*

Table 2 Hardware implementation of medical image denoising

Refs.	Algorithms/method	Platforms	Target applications
[14]	advanced wavelet domain-noise filtering	XC2V6000-5	video denoising
[15]	SVD	XCV2000E-6	image denoising
[16]	Cohen Daubechies-CDF 5/3, 9/7	XCV300	image processing
[17]	structured-oriented noise estimation	XC2V4000	PAL video sequence

15 000 gates are utilised, at 2.185 MHz maximum clock speed and 24 mW power consumption.

A hardware architecture for video noise estimation is discussed in [17] by Lapalme *et al.* The aim of this architecture is to process two consecutive fields simultaneously by acquiring data within two modules. This concept implies a 'ping-pong' structure and refers to the back and forth processing scheme to enable parallel processing. The advantages of this architecture are the design which is uniformly pipelined and digital clock managers (DCMs) are used to accelerate the sorting mechanism's clock frequency, as well as maximise the parallelism of arithmetic operations to achieve real-time requirements. In terms of hardware implementation strategy, the use of a generic VHDL contributes to a scalable architecture that can accommodate different filter sizes. Moreover, the experimental results reveal that the noise variance generated was proven to be accurate in comparison with the original software simulation findings.

Table 2 presents a summary for hardware implementation of image and video denoising; and classification in terms of algorithms or methods implemented, FPGA devices and target applications. Design strategies that have been manipulated such as systolic array, FPGA resources optimisation and generic design style provide new directions for future work to be carried out on hardware implementation of image denoising using FRAT.

2 Mathematical background

This section presents the mathematical background for Radon transform (RT) algorithms.

2.1 Radon transform

Mathematically, the RT in two dimensions is the integral transform comprising of a function over straight lines. If a straight line is represented parametrically by (1)

$$s = x\cos\theta + y\sin\theta \tag{1}$$

where s is the shortest distance from the straight line to the origin and θ is the angle which the line makes with the y-axis, then the RT function $R[f](\theta, s)$ can be given as follows

$$R[f](\theta, s) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x, y)\delta(x\cos\theta + y\sin\theta - s)\,\mathrm{d}x\,\mathrm{d}y \tag{2}$$

where δ is the Dirac impulse function. Based upon (1) and (2), the inverse Radon transform (IRT) is given in (3), which is used for image reconstruction.

$$f(x, y) = \int_0^{2\pi} R[f](\alpha, x\cos\alpha + y\sin\alpha)\,\mathrm{d}\alpha \tag{3}$$

2.2 Finite Radon transform

The FRAT is defined as the summation of image pixels over a certain set of lines. Those lines are defined in a finite geometry in a similar way to the lines for continuous RT in Euclidean geometry.

Consider a cyclic group Z_p denoted by $Z_p = (0, 1, \ldots, p-1)$ where p is a prime number. Let the finite grid Z_p^2 be defined as the Cartesian product $Z_p \times Z_p$. This finite grid has $(p+1)$ non-trivial sub-groups, given by

$$L_{k,l} = (i, j){:}j = (ki + l)(\bmod p),\ \ i \in Z_p,\ \ 0 \le k\langle p \tag{4a}$$

$$L_{p,l} = (l, j){:}j| \in Z_p \tag{4b}$$

where each sub-group $L_{k,l}$, is the set of points that define a line on the lattice Z_p. The Radon projection of the function f on the finite grid Z_p^2 is given by

$$r_k[l] = \mathrm{FRAT}_f(k, l) = \frac{1}{\sqrt{p}}\left(\sum_{(i,j)\in L_{k,l}} f[i, j]\right) \tag{5}$$

From (4) and (5), it can be seen that the function f is treated as a periodic function. Therefore the digital representation of the line displays a 'wrap around' effect. Analogous to the continuous case, as in Euclidean geometry, any two lines intersect at only one point in the finite grid Z_p^2. Hence, the inverse transform, the finite back projection (FBP) is given by

$$\mathrm{FBP}_r(i, j) = \frac{1}{\sqrt{p}}\left(\sum_{(k,l)\in P_{i,j}} r_k[l],\ (i, j) \in Z_p^2\right) \tag{6}$$

where $P_{i,j} = (k, l){:}l = (j - ki)(\bmod p),\ k| \in Z_p. \cup (p, i)$

Substituting (5) into (6), (7) and (8) proves that the FBP provides a perfect inversion for the FRAT [8]. Also, the algorithms for the FBP and FRAT are synonymous. As a result, the same architecture can be used to implement both the forward and inverse transforms.

$$\mathrm{FBP}_r(i, j) = \frac{1}{p}\left(\sum_{(k,l)\in P_{i,j}}\ \sum_{(i',j')\in L_{k,l}} f[i', j']\right) \tag{7}$$

$$\mathrm{FBP}_r(i, j) = \frac{1}{p}\left(\sum_{(i',j')\in Z^p} f[i', j'] + pf[i, j]\right) = f[i, j] \tag{8}$$

Computing the Kth Radon projection, that is, the Kth row of the array, all pixels of the original image need to be passed once and use p histograms: one for every pixel in the row. At the end, all p histogrammed values are divided by p to obtain the average values.

2.3 Image denoising

Medical acquisition technologies and systems introduce noise and artefacts in the images that should be attenuated by the denoising algorithms [12], likewise, the denoising process

doi: 10.1049/iet-spr.2011.0392

must ensure that the anatomical details are retained for critical computer-aided analysis of the images.

Mathematically, image denoising can be summarised as follows. Let $A(i, j)$ be the noise-free image and $B(i, j)$ the image contaminated with independent Gaussian noise $Z(i, j)$

$$B(i, j) = A(i, j) + \sigma Z(i, j) \tag{9}$$

where $Z(i, j)$ has normal distribution $N(0, 1)$. The problem is to estimate the desired signal as accurately as possible according to some criteria. In the Radon domain, the problem can be formulated as follows

$$Y(i, j) = R(i, j) + N(i, j) \tag{10}$$

where $Y(i, j)$ is noisy Radon coefficient, $R(i, j)$ is true coefficient and $N(i, j)$ is the noise.

2.4 Proposed architectures

Proposed system applications and an overview of the architecture with different design strategies are described in the following section.

2.5 Proposed denoising system

Fig. 2 illustrates an overview of our proposed 3D medical image denoising system based on FRAT. In each block, buffers have been used for storing intermediate results to be processed.

It is well known that noise in medical images results in low image quality and yet limits diagnostic effectiveness. In real implementation, the proposed method can be modelled as follows

$$Y = \mathrm{IRT}(\mathrm{RT}(X + N)) \tag{11}$$

X, N and Y refer to the raw medical image, the noise introduced by the equipment and the output image, respectively. From (11), it is implied that

$$Y \simeq X + \mathrm{IRT}(N) \tag{12}$$

and the noise in the medical image is approximately

$$\mathrm{IRT}(N) \simeq Y - X \tag{13}$$

Therefore noise reduction for medical images is significantly vital for the pre-processing stage before performing compression. Generally, three steps are involved in image denoising: calculate the transform of the noisy signal, modify the noisy radon coefficients according to the specific rule and compute the inverse transform using the modified coefficients. To be more specific, the denoising algorithm operates as follows:

Step 1: Adding a Gaussian white noise to the image, then applying the FRAT to the noisy image;
Step 2: Calculating the threshold and thresholding the FRAT coefficients with universal thresholding; and

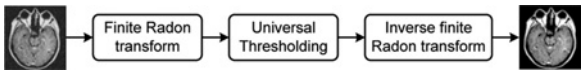

Fig. 2 *Proposed denoising system*

Step 3: Inverse transform of the thresholded coefficients.

It is worth noting that thresholding plays a significant role in the denoising process [18] and finding an optimum threshold is a tedious process. A universal threshold as proposed by Donoho and Johnstone in [19] is defined as follows

$$\lambda = \sigma_n \sqrt{\log_2 N} \tag{14}$$

where N refers to the image size and σ is the noise standard deviation.

3 Design method

To ease the process of transforming a MATLAB [20] floating point design into a hardware module, Xilinx DSP tool software is used for rapid prototyping of algorithms in MATLAB into hardware [21]. The main feature of the Xilinx DSP tool can be summarised as follows:

- A synthesisable register transfer level (RTL) design can be obtained from the floating point M-code;
- A set of test-benches can be automatically generated; and
- Capability to invoke hardware description language (HDL) simulation, synthesis and implementation tools.

Verification in each stage is very significant. The tool verifies the generated module in each step to be as true as the previous one, or to be subjectively acceptable with a minor difference during the conversion from floating point design to fixed point [21, 22]. There are two main parts in the M-code: a script and function file [21]. In addition, there are three functions of script files. It creates stimuli, feeds the stimuli to the function in a streaming loop and verifies the output from the function. On top of that, the script file also serves as a source file for future test-bench auto generation. Furthermore, the function file comprises the actual function to be translated into HDL and it is written as an ordinary MATLAB function with an interface of input and output variables. In this study, the Xilinx DSP tool has been selected, since it can be automatically converted from high-level languages to RTL HDL and even directly to an FPGA configuration bitstream [22]. This feature is important for reducing the design cycle as well as allowing more optimisation to be carried out on the algorithmic and architectural levels.

The high-level description of the denoising algorithms for FRAT, inverse FRAT and thresholding are synthesised to generate a device-specific HDL RTL description of a hardware implementation targeting of an FPGA. The RTL description is co-simulated using Modelsim to verify a low-level functionality of the system and generate the switching activities for power analysis. To perform a more precise estimation of maximum frequency, area as well as power, the synthesis results are then imported in the ISE tool.

4 Proposed architectures

Fig. 3 shows the reference architecture [3, 7] for the FRAT based on the FRAT's pseudo-code. This reference architecture has been used as an illustration of the FRAT algorithm to explain the concept. To exploit the hardware resources available, the operations of the various counters used to track the addresses of the output vectors are

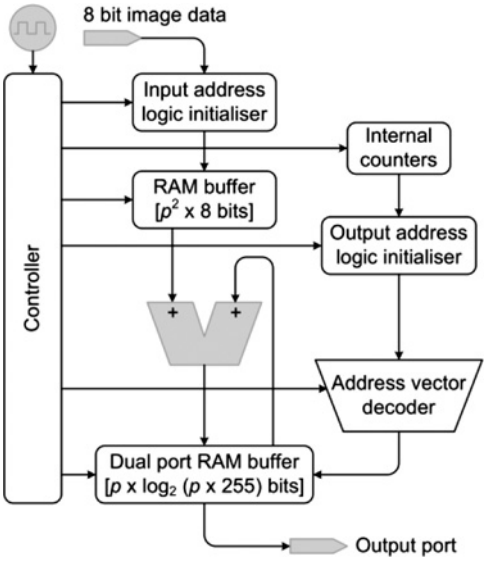

Fig. 3 *Proposed reference architecture for FRAT [3, 7]*

parallelised and pipelined. This has been carried out by changing rollover conditions and count limits suitably.

The number of counters required remains the same, and only the triggering conditions, order and reset logic are modified suitably. It must be highlighted that although the algorithm is still serial and cycles through $p \times (p+1)$ iterations, the number of steps in the algorithm have been reduced, thereby improving latency. The architecture has serial I/Os and a serial core. The total latency of the core is $O\ (p^2 \times (p+1))$. The input section consists of a one-dimensional (1-D) random access memory (RAM) of width 8 bits and a depth of p^2.

Although each input image block is a square tile of side p, buffering it in a 1-D RAM reduces the computational complexity of the control logic associated with data access. This is because, a two-dimensional RAM is implemented on the FPGA as a number of 1-D RAMs and uses additional multiplexing logic to dereference the address locations. The FRAT operation requires reading and writing from the same memory location within a single clock pulse. This is compactly and effectively implemented using a dual ported RAM at the output section instead of an array-based buffer. The output buffer is a 1-D dual port RAM of width $\log_2 p.256$ and depth p. Only a single FRAT vector is buffered and the final values are written to the output port in serial fashion at the end of each iteration. At the end of $(p+1)$ iterations, the entire image block is transformed to the FRAT domain.

Based on the FRAT architecture, three design strategies have been proposed as shown in Figs. 4*a*–*c*, with '*R*', '*E*' and '*W*' referring to 'Read', 'Enable' and 'Write' processes, respectively.

For the sequential fashion, the following modes involve:

Mode 1: Transfer data, a block of $p \times p$ pixels, pixel by pixel;
Mode 2: Compute FRAT; and
Mode 3: Transfer results, a block of $p \times p$ pixels, pixel by pixel.

On the other hand, the pipelined implementation modes are as follows:

Mode 1: Transfer data in pipelined fashion, read a block of $p \times p$ pixels, 7 column, column by column (7 pixels at a time) simultaneously write a block of $p+p \times p$ pixels, 7 column, column by column (7 pixels at a time); and
Mode 2: Compute FRAT.

In the case of the BRAM-based method, the processes involved can be described as follows:

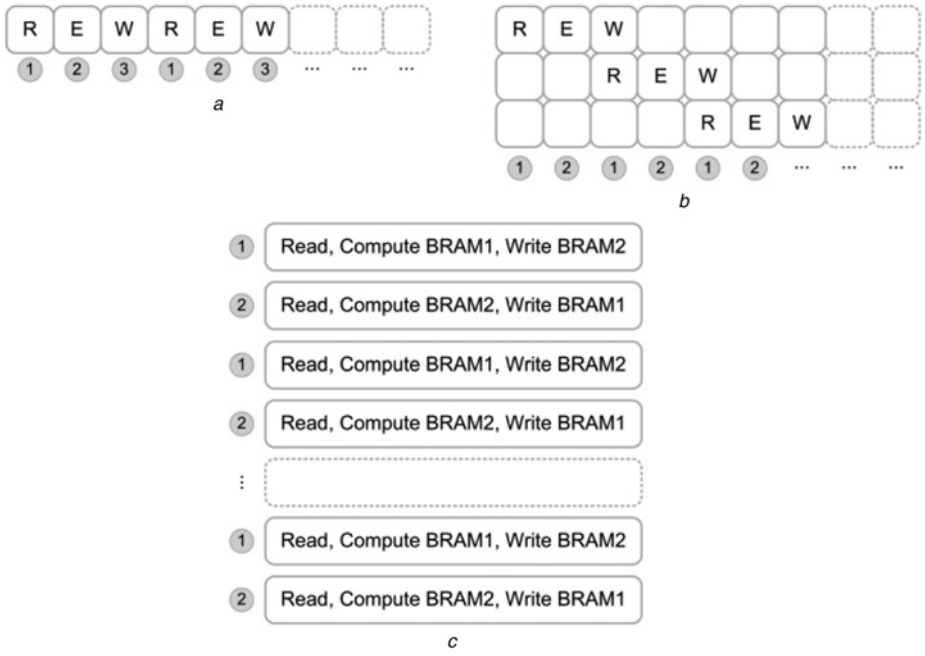

Fig. 4 *Implementation strategies*

a Sequential
b Pipelined
c BRAM-based method

doi: 10.1049/iet-spr.2011.0392

Table 3 PSNR quantitative results of noisy image with Gaussian white noise and MRI image

Σ	Noisy, dB	Denoising, dB		
		Block sizes, *p*		
		7	17	31
0.01	31.15	43.60	39.51	36.48
0.02	30.63	41.68	37.75	34.67
0.04	30.30	39.34	35.47	32.38
0.08	30.06	36.98	32.95	29.89
0.16	29.91	34.48	30.44	27.39

Mode 1: Compute the FRAT block in BRAM 1, and write the computed FRAT block in BRAM 2; and
Mode 2: Compute the FRAT block in BRAM 2, and write the computed FRAT block in BRAM 1.

The proposed architectures for the FRAT algorithm are applicable to executing the process of inverse FRAT because of the similarities between these two algorithms, FRAT and inverse FRAT. FRAT, denoising and inverse FRAT are described as functions. The FRAT function takes an image block as an input argument and its output is used as an input argument for the denoising function. Finally, the inverse FRAT generates a filtered image block. In the following section, the implementation results of the FRAT algorithm with a comparison with existing similar architectures; and the implementation of the complete denoising systems will be given.

5 Results and analysis

Three types of images [23] have been used for software simulation and hardware implementation: medical resonance imaging (MRI) scan of human brain (940 × 940), chest body computerised tomography (CT) (128 × 128) and positron emission tomography (PET) scan of normal human brain (109 × 109). To evaluate the quality of the processed images and to demonstrate the effectiveness of FRAT, PSNR has been calculated to quantitatively estimate noise suppression.

5.1 Medical image denoising

To analyse the effectiveness of FRAT in medical image denoising, a Gaussian white noise with mean μ zero and various standard deviations σ has been added to the experimental images.

By utilising FRAT in medical image noise reduction, the results obtained have shown promising results. The denoising results obtained reveal that FRAT implementation is effective in reducing Gaussian noise. Table 3 shows quantitative results for MRI images, whereas Figs. 5*a*–*f* illustrate the significant achievement of 13.25 and 22.6% denoising for the MRI image using FRAT.

5.2 Software simulation

Figs. 6*a*–*l* show FRAT domain visualisation of MRI and PET slices. The averaging impact of FRAT on image blockiness in the transform domain can be observed as p increases.

It is worth noting that the FBP is a mathematically perfect inversion for FRAT and the PSNR depends only on the accuracy required. The truncation or rounding step that follows the FRAT determines the PSNR values. As it is usually used as a sub-block in other transforms such as FRIT and curvelets and is followed by a wavelet stage in these transforms, the rounding or truncation process can easily be incorporated along with the wavelet block with no extra computational effort by suitably modifying the wavelet coefficients Table 4.

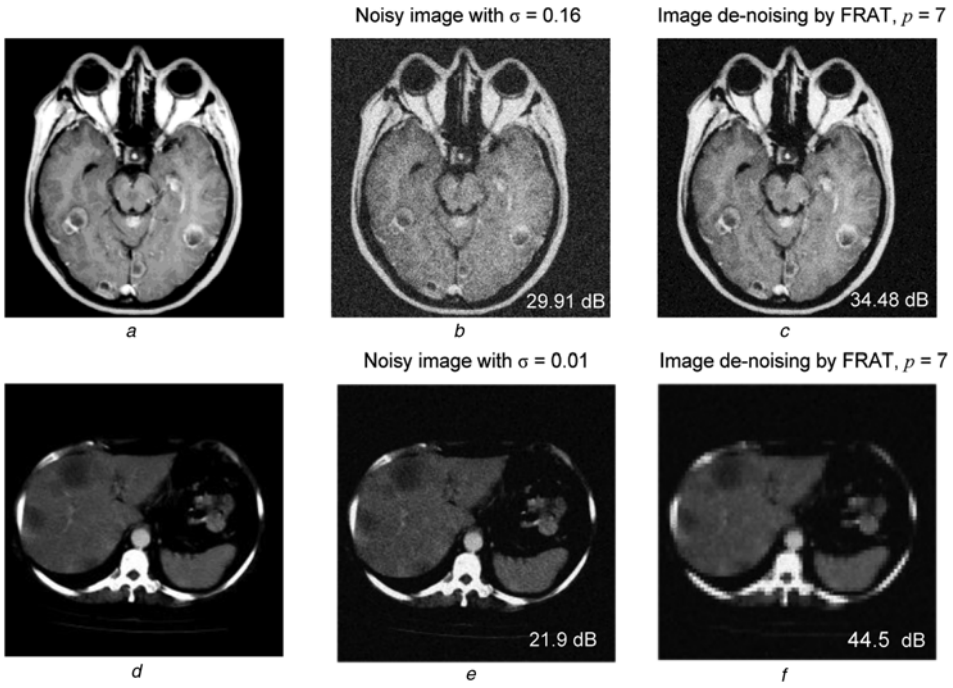

Fig. 5 *Gaussian noise reduction experimental results on MRI, CT images*
a and d Original
b and e Noisy
c and f Denoising

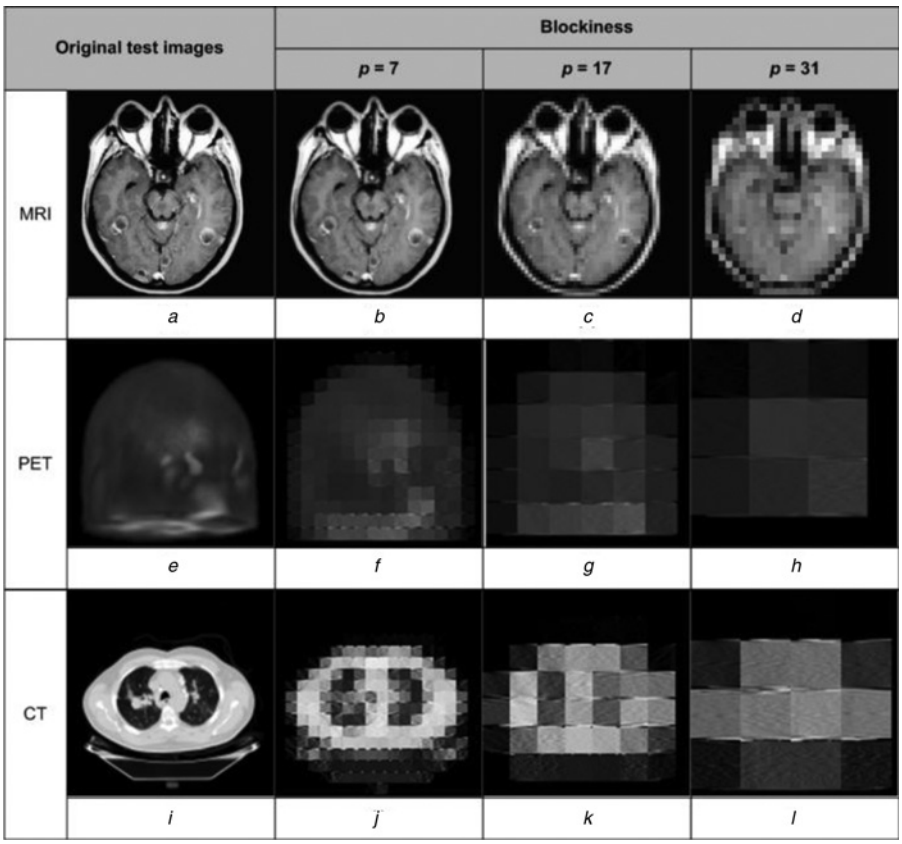

Fig. 6 *FRAT domain visualisation of MRI, PET and CT slices*

a, *e*, *i* Original images
b–d, *f–h*, *j–k* Blockiness images

However, to illustrate the effect of bit-width limitations on the PSNR, Fig. 7 shows the relationship of the PSNR values for the reconstructed medical images with various block sizes p. The results obtained exhibit that the PSNR of the reconstructed image drops by 7.93, 21.70 and 21.80 dB for MRI, CT and PET, respectively, when the block size increases from $p = 7$ to 31. This is because as p increases, the rounding error becomes more significant. Using a divider with greater precision can reduce the rounding error.

5.3 Hardware implementation

For all the three cases of hardware implementation: sequential, pipelined and BRAM-based method, pseudo-codes have been implemented in MATLAB and the Xilinx DSP tool has been used for architecture development and synthesis exploration.

The designs have been implemented on the Virtex-5 (XC5VLX110 T) FPGA device. As the prime aim of this paper is to examine the best hardware implementation applied for medical image denoising, results for both medical image denoising as well as the software simulation justify hardware implementation with $p = 7$. Comparison of performance metrics for the proposed FRAT architectures with the existing work is presented in Table 5.

Table 4 Analysis of PSNR with different block sizes (p)

Block sizes, p	MRI	PET	CT
5	51.94	52.20	41.90
7	49.76	50.20	41.70
11	47.17	41.10	36.70
17	45.36	38.20	29.80
31	41.83	28.50	19.90

The implementation results of the proposed architectures show the estimated number of occupied slices. In the pipelined architecture, all the loops are unrolled which explains the augmentation of the occupied slices. Even though the number of occupied slices is more important in a pipelined architecture compared with the others, this type of implementation has been adopted because of its high performance capabilities.

To the best of our knowledge, there is no complete hardware implementation of the denoising system based on FRAT. The existing software solutions generally present the quality of filtering without mentioning the utilisation of resources as well as other important parameters such as power consumption. Hence, Table 6 gives the implementation results for the pipelined architecture of the FRAT compared with the sequential and the overall system.

As can be expected, the parallel with loops unrolled architecture of FRAT outperforms the sequential architecture with minimum power. On top of that, the performances of the overall system composed of FRAT, filtering and inverse FRAT are not degraded compared with FRAT. The system can filter a 7×7 block in 0.344 μs, which corresponds to 142.5 million pixels per second (MPPS).

In this study, Xpower tool from the Xilinx design suite has been fully utilised to estimate power consumption. The results obtained also reveal the quiescent power of these three architectures with 1044 mW which depends mainly on the

doi: 10.1049/iet-spr.2011.0392

Table 5 Comparison of performance with the existing architectures of FRAT for the case $p = 7$

Type	Platform	Design	*F*, MHz	*T* (MPPS)	*A* (Slices)
sequential	Virtex-E	[3]	94.46	45.01	245
	Virtex-II	[7]	69.00	6.90	345
		[3]	79.973	37.32	215
		[5]	100.1	9.87	159
		[6]:A1	112.87	11.13	198
		[6]:A2	67.3	6.64	131
	Virtex-5	proposed	238.10	3.5	110 + 1 BRAM
pipelined	Virtex-5	proposed (1)	200	200	1687
BRAM-based	Virtex-5	proposed	188.90	188	637+ 4 BRAMs

Note: (1) Loops unrolled

Table 6 Comparison of different FRAT architectures and system architectures

Parameter (unit)	Proposed architecture		
	Sequential bit accurate (FRAT)	Loop unrolled (FRAT)	Loop unrolled (System)
min. period, ns	4.20	4.97	4.98
max. frequency (MHz)	238.10	200	200
*latency, cycles	3297	49	69
*latency, time μs	13 847	0.244	0.344
throughput, MPPS	3.5	200.8	142.5
total power, mW	1122	1,301	1,608
total slices	110	1,687	4,627
power.time, W.μs	15.5	0.32	0.55

Note: *Execution for a 7 × 7 block of data

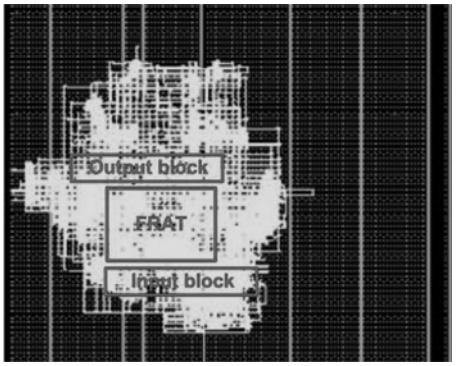

Fig. 7 *Chip layout for sequential implementation*

target device. The proposed architecture also exhibits 0.32 μJ of energy, which can also be considered as efficient energy consumption compared with the sequential architecture.

The results achieved for hardware implementation demonstrate various trade-offs with sequential and pipelined descriptions yielding better achievement for maximum frequency (F) and throughput (T), respectively. Moreover, the BRAM-based method also reveals less area (A) occupied and better maximum frequency. To visualise the design and implementation of the FRAT, Fig. 7 illustrates the chip layout for sequential implementation on XC5VLX110 T FPGA device. Fig. 8 illustrates the chip

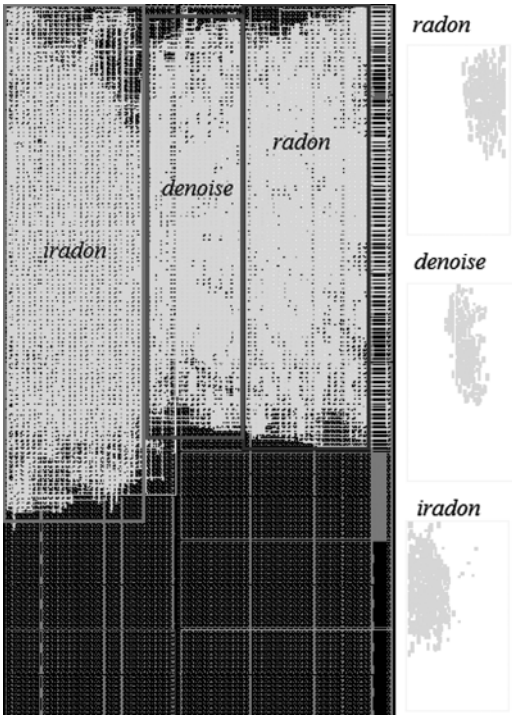

Fig. 8 *Chip layout for parallel implementation of the full denoising system (foot prints for radon, denoise and iradon modules depicted separately on the right side)*

layout for parallel implementation on the same device with footprints of 'radon', 'denoise' and 'iradon' sub-modules, depicted separately on the right side.

A detailed comparison for both hardware implementation and software simulation with test medical images has been carried out. As shown in Table 7, software simulation achieved better PSNR over hardware implementation with the percentage difference being 12.92, 21.47 and 33.09% for $p = 7$, 17 and 31, respectively. This is due to the use of

Table 7 Comparison of PSNR Values for CT images

Implementation	PSNR, dB		
	Block sizes, *p*		
	7	17	31
Hardware	46.30	38.13	30.30
Software	53.71	48.56	45.29

a floating point in MATLAB, which yields better PSNR values compared with a fixed point model in hardware implementation.

6 Conclusion

An FPGA-based architecture with three different design strategies has been proposed for efficient implementation of FRAT. An in-depth analysis of the results obtained with various medical imaging modalities using the proposed denoising system has been conducted. Image denoising implementation using the FRAT exhibits a significant achievement to reduce Gaussian white noise in medical images. An evaluation of the implementation results outperforms the existing architectures in terms of maximum frequency, throughput and area. The hardware implementation of the overall denoising system including FRAT, filtering and inverse FRAT can process up to 142.5 MPPS.

7 Acknowledgments

The authors would like to thank the British Council, the French programme Alliance Egide for funding this research work and the Ministry of Higher Education, Malaysia and Universiti Tun Hussein Onn Malaysia (UTHM) for funding Mr. Ahmad's PhD study.

8 References

1 Chen, G.Y., Kgl, B.: 'Image denoising with complex ridgelets', *Pattern Recognit.*, 2007, **40**, (2), pp. 578–585
2 Zhang, D., Nishimura, T.: 'Medical image noise reduction using Radon transform and Walsh list in Laplacian pyramid domain'. ISCE'09 IEEE 13th Int. Symp. on Consumer Electronics, 25–28 May 2009, pp. 756–760
3 Chandrasekaran, S., Amira, A., Minghua, S., Bermak, A.: 'An efficient VLSI architecture and FPGA implementation of the finite ridgelet transform', *J. Real-Time Image Process.*, 2008, **3**, (3), pp. 183–193
4 Wisinger, J., Mahapatra, R.: 'FPGA based image processing with the curvelet transform'. TX, Technical report TR-CS-2003-01-0, Texas A & M University, 2003
5 Rahman, C.A., Badawy, W.: 'Architectures the finite Radon transform', *IEE Electron. Lett.*, 2004, **40**, (15), pp. 931–932
6 Uzun, I.S., Amira, A.: 'Design and FPGA implementation of finite ridgelet transform'. Proc. IEEE Int. Symp. on Circuits and Systems (ISCAS), May 2005, vol. 6, pp. 5826–5829
7 Chandrasekaran, S., Amira, A.: 'High speed/low power architectures for the finite Radon transform'. Proc. Int. Conf. on Field Programmable Logic and Applications, August 2005, pp. 450–455
8 Matus, F., Flusser, J.: 'Image representation via a finite Radon transform', *IEEE Trans. Pattern Anal. Mach. Intell.*, 1993, **15**, pp. 996–1006
9 Zha, D., Qiu, T.: 'A new algorithm for shot noise removal in medical ultrasound images based on alpha-stable model', *Int. J. Adapt. Control Signal Process.*, 2006, **20**, pp. 251–263
10 Liu, Y.X., Peng, Y.H., Siu, W.C.: 'Energy-based adaptive transform scheme in the DPRT domain and its application to image denoising', *Signal Process.*, 2009, **89**, (1), pp. 31–44
11 Guo, L., Au, O.C., Ma, M., Liang, Z.: 'Fast multi-hypothesis motion compensated filter for video denoising', *J. Signal Process. Syst.*, 2010, **60**, (3), pp. 273–290
12 Sanches, J., Nascimento, J., Marques, J.: 'Medical image noise reduction using the Sylvester Lyapunov equation', *IEEE Trans. Image Process.*, 2008, **17**, (9), pp. 1522–1539
13 Bao, P., Zhang, L.: 'Noise reduction for magnetic resonance images via adaptive multiscale products thresholding', *IEEE Trans. Med. Imag.*, 2003, **22**, (9), pp. 1089–1099
14 Katona, M., Pižurica, A., Teslíc, N., Kovacevic, V., Philips, W.: 'A real-time wavelet-domain video denoising implementation in FPGA', *EURASIP J. Embedded Syst.*, 2006, **2006**, (1), pp. 1–12
15 Ahmedsaid, A., Amira, A.: 'Accelerating SVD on reconfigurable hardware for image denoising'. ICIP'04 Int. Conf. on Image Processing, 2004, vol. 1, pp. 259–262
16 Savakis, A.E., Carbone, R.: 'Discrete wavelet transform core for image processing applications', *Real-Time Imaging IX*, 2005, **5671**, (1), pp. 142–151
17 Lapalme, F.X., Amer, A., Wang, C.: 'FPGA architecture for real-time video noise estimation'. IEEE Int. Conf. on Image Processing, 2006, pp. 3257–3260
18 Mohideen, S.K., Perumal, A., Sathik, M.M.: 'Image de-noising using discrete wavelet transform', *Int. J. Comput. Sci. Netw. Secur.*, 2008, **8**, (1), pp. 213–216
19 Donoho, D.L., Johnstone, I.M.: 'Adapting to unknown smoothness via wavelet shrinkage', *J. Am. Stat. Assoc.*, 1995, **90**, (432), pp. 1200–1224
20 'Accessed 1 May 2010' http://www.mathworks.com
21 'AccelDSP Synthesis Tool User Guide'. *Xilinx Inc.*, Technical report UG634 (v11.4), December 2009
22 Yu, G., Vladimirova, T., Wu, X., Sweeting, M.: 'A new high-level reconfigurable lossless image compression system for space applications'. AHS'08. NASA/ESA Conf. on Adaptive Hardware and Systems, June 2008, pp. 183–190
23 'Accessed 10 February 2009' http://www.rsna.org

doi: 10.1049/iet-spr.2011.0392