Design of a Gabor Filter HW Accelerator for Applications in Medical Imaging

Gian Domenico Licciardo, *Senior Member, IEEE*, Carmine Cappetta, *Student Member, IEEE*, and Luigi Di Benedetto, *Member, IEEE*

***Abstract*— The Gabor filter (GF) has been proved to show good spatial frequency and position selectivity, which makes it a very suitable solution for visual search, object recognition, and, in general, multimedia processing applications. GFs prove useful also in the processing of medical imaging to improve part of the several filtering operations for their enhancement, denoising, and mitigation of artifact issues. However, the good performances of GFs are compensated by a hardware complexity that traduces in a large amount of mapped physical resources. This paper presents three different designs of a GF, showing different tradeoffs between accuracy, area, power, and timing. From the comparative study, it is possible to highlight the strength points of each one and choose the best design. The designs have been targeted to a Xilinx field-programmable gate array (FPGA) platform and synthesized to 90-nm CMOS standard cells. FPGA implementations achieve a maximum operating frequency among the different designs of 179 MHz, while 350 MHz is obtained from CMOS synthesis. Therefore, 86 and 168 full-HD (1920 × 1080) f/s could be processed, with FPGA and std_cell implementations, respectively. In order to meet space constraints, several considerations are proposed to achieve an optimization in terms of power consumption, while still ensuring real-time performances.**

***Index Terms*— Accelerators, field-programmable gate array (FPGA), Gabor filters (GFs), hardware (HW), visual search (VS).**

I. Introduction

MEDICAL diagnostic techniques underwent significant improvements in the last years, due to the availability of new electronic apparatus, e.g., X-ray computed tomography (CT) scan and magnetic resonance (MR), based on the noninvasive detection of anomalies from acquired images, as well as processing techniques developed *ad hoc* for computer-aided diagnosis (CAD). In particular, the development of methods for the automatic or assisted identification of specific features into images is fundamental for good disease detections and accurate diagnostic results [1], [2]. However, the recurrent presence of artifacts, blurring effects, and, in general, noise in the acquired images introduces ambiguities in the interpretation of the results [3], which could result in wrong diagnosis. Among the different processing methods for the mitigation of the above problems, in the medical context several papers have shown that Gabor filter (GF) is capable of very good performances when it is directly used for feature extraction, as well as for image enhancement during preprocessing steps. With reference to the recent literature, in [4], 2-D GF has been used to extract tumor-related features from mammography; in [5], a bank of GFs has been used to implement a new feature extraction method from medical images; in [6], a bank of GFs, each tuned with a different orientation, is used for medical image segmentation and verified on MRI and CT samples; in [7], GF has been used for the quality enhancement of X-ray images; and in [8], a GF bank has been used for the preprocessing of images for the early diagnosis of lung cancer. More recently, in [9], a new technique based on log-GF has been presented to reduce color distortion in PET and MR images, while in [10], a novel 3-D Gabor wavelet transformation is introduced for revealing the instrument voxels in the volume of ultrasound acquisitions.

The popularity of GF, not only in medical applications, but also in edge and corner detections, image segmentation, gait analysis, visual search (VS), and even speech recognition [11]–[14], is mainly due to the peculiar characteristics of the filter to well describe the behavior of mammalian visual cortex cells [15] and minimize the joint uncertainty for both frequency and spatial position [16]. As it will be shown in Section II, its good characteristics are compensated by a high computational complexity, mainly related to the infinite length impulse response and the accuracy that increases with the number of orientations to compute. This complexity is much more relevant in medical image processing, where very high resolutions are required.

The existing literature [11]–[17] shows that the software (SW) approach is by far the preferred way of CAD systems to deal with GF complexity, thanks to the high flexibility to modify, adapt, and reconfigure single SW modules or entire processing pipelines, while preserving the adequate accuracy for medical uses. However, the need for high-performance, general-purpose processors to obtain acceptable throughputs prevents integration and the deriving opportunities. Indeed, the availability of dedicated hardware (HW) solutions with a favorable area–power–delay (APD) tradeoff have demonstrated unique performances in applications devoted to data preprocessing and data transfer, optimizing the operators involved in the filtering operations [18]–[21]. In our case, devoted and resource-saving HW enables the design of

Manuscript received September 27, 2017; revised January 12, 2018 and March 19, 2018; accepted March 20, 2018. Recommended for publication by Associate Editor D. De Venuto upon evaluation of reviewers' comments. *(Corresponding author: Carmine Cappetta.)*

The authors are with the Department of Industrial Engineering, University of Salerno, 84084 Fisciano, Italy (e-mail: gdlicciardo@unisa.it; ccappetta@unisa.it; ldibenedetto@unisa.it).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCPMT.2018.2818947

high-performance DSPs [22] as well as smart sensors, where image sensors are tightly coupled (e.g., in a package) with the processing stages, in order to be used in small or handheld devices, like those for minimally invasive surgery [23] and the small apparatus for portable and wearable auto diagnostic systems [24], [25].

To the best of our knowledge, the existing literature proposes a number of Gabor-like HW designs that implement simplified/reduced kernels, adapted to specific purposes but far from exploiting all the GF features needed for medical applications. The state-of-the-art performances are reported in [26] and [27] where a cellular neural network has been set up to implement Gabor-like filters having simplified coefficients, with only one 3×3 scale and only one orientation that is insufficient for CAD.

This paper proposes a new application-specific image processor (ASIP) that implements the full features of GF so that it can be successfully used for CAD applications with optimal APD tradeoff. Since there are several GF parameters to tune the computational complexity and the amount of mapped physical resources to target platforms with different capabilities, three designs have been proposed by varying the number of orientations and enabling the possibility to share part of the datapath circuitry. The proposed ASIP has been implemented to field-programmable gate array (FPGA) platforms and synthesized with TSMC CMOS 90-nm standard cells (std_cell). The best-case maximum path delay is 5.6 ns, allowing to process a full-HD frame in 11.61 ms on a Xilinx Virtex 7 FPGA [28]. Synthesis with std_cell achieves a best-case maximum path delay of 2.86 ns, therefore 5.93 ms to process a full-HD image.

The remainder of this paper is organized as follows. Section II highlights the theory behind GFs. Section III provides the considerations bringing to the choice of GFs parameters. Section IV reports the numerical remarks and considerations underlying the design choices. Section V presents the HW designs, examining the various modules composing them. Section VI provides the results for the various implementations and compares them to the recent literature. Finally, Section VII provides the conclusion.

II. Gabor Filters Theory

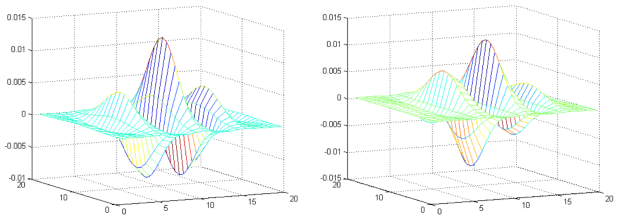
Two-dimensional (2-D) Gabor functions are a class of functions obtained multiplying a generic 2-D Gaussian function for a complex exponential. The obtained function could be written in the form [11]

$$\psi\ (x, y; f_0, \theta) = \frac{f_0^2}{\pi\,\gamma\,\eta} e^{-\left\{\frac{f_0^2}{\gamma^2}x'^2+\frac{f_0^2}{\eta^2}y'^2\right\}} e^{j2\pi f_0 x'} \qquad (1)$$

with

$$\begin{cases} x' = x\cos\theta + y\sin\theta \\ y' = -x\sin\theta + y\cos\theta \end{cases}$$

where x and y represent the spatial coordinates of the considered filter, f_0 central frequency of the considered GF, θ rotation angle for the particular orientation of the filter, γ sharpness of the Gaussian along the major axis, and η sharpness of the Gaussian along the minor one. An example



(a) (b)

Fig. 1. (a) Real part and (b) imaginary part of GF with $f_0 = 0.2$ cycle/pixels, $\theta = 0$, $\gamma = 1$, and $\eta = 1$.

of Gabor function for $f_0 = 0.2$ cycles/pixels, $\theta = 0$, $\gamma = 1$, and $\eta = 1$ is shown in Fig. 1, where it has been decomposed in its real and imaginary parts. The real part of the Gabor function is an even function, while its imaginary part is an odd one. This makes it possible to exploit the separation of the 2-D kernels along their orthogonal directions, with a consequent reduction in the computational complexity of the filtering operation along the directions $\theta_s = \{0, \pi/2\}$, as it will be shown in the following.

As demonstrated in [11], features obtained via filtering with Gabor functions are stable in terms of image rotation, scaling in frequency, and translation, which also allows us to state that Gabor features are robust in terms of noise rejection, image distortion, and objects deformation. Invariance to image rotations is obtained by setting θ since the filter response for rotated images could be obtained via filtering the original nonrotated image with the rotated Gabor kernel.

Further considerations developed in [12] and [29] show that, in edge detection applications, it is possible to obtain robust results using only the imaginary part of the filtered data. Since the input image is always represented by real values, it is possible to use only the imaginary part of the GF

$$G(x, y, f_0, \theta) = \frac{f_0^2}{\pi\,\gamma\,\eta} e^{-\left\{\frac{f_0^2}{\gamma^2}x'^2+\frac{f_0^2}{\eta^2}y'^2\right\}} \sin\ (2\pi f_0 x') \qquad (2)$$

avoiding modulus calculations, and thus achieving further reduction of the time required to perform the entire operation and of the number of operations associated.

III. Implemented Filter and Parameters Choice

Edge detection is one of the primary uses of GFs in CAD. In order to implement robust and scale-invariant edge detection with high sensitivity, a multiscale approach is needed, where different samples of the same image with scaled resolutions are processed, which causes an important increase of the physical resources to map for the HW implementation of the ASIP [12], [30]. However, an important simplification comes from avoiding the linear summation of the Gabor results from different orientations on each scale, and considering only the maximum value along the different orientations for each pixel location [29]. In such a way, the use of a bank of GFs having different central frequencies has been avoided in our design, in favor of a single multiscale GF, where the original Luma component of the image is filtered with a defined set of orientations at each scale. In this way, a set of $N_S \times N_O$ candidate

features is obtained, where N_S is the number of considered scales and N_O the number of orientations for each scale. From the set of candidates obtained for each pixel location, only the maximum is chosen as proposed feature to compose the final resulting image. Such an approach reduces also the mapped resources for the datapath, since negative results are discarded, according to a maximum selection method.

In the following, the parameters determining the GF function are examined and considerations regarding the chosen number of scales are reported in order to establish an optimal set of parameters for the proposed designs.

A. Filter Central Frequency f_0

The central spatial frequency of the GF f_0 is always in the range [0, 0.5] cycle/pixels since the maximum frequency to be considered, $f_N = 0.5$ cycle/pixels, is given by the Nyquist theorem. However, central frequencies below 0.1 cycle/pixels could cause a degradation in the feature accuracy due to higher blurring in the resulting image and to the impossibility of following sharp edges. On the other side, frequencies above 0.2 cycle/pixels may require filters with narrower bandwidths, requiring an unwanted oversampling of the kernel with respect to the resolution grid of the input image [13]. In our designs, $f_0 = 0.2$ cycle/pixels has been chosen to satisfy the above constraints while minimizing the blurring effect.

B. Set of Orientations θ

The number of considered orientations is one of the primary concerns in GF applications. This is due to several considerations regarding both computational complexity and resource usage. Table I shows that the number of arithmetic operators increases linearly with N_S when the number of orientation is two (2Or), thanks to the separability of the GF kernels at $\theta = 0$ and $\theta = \pi/2$, as shown ahead. Since intermediate orientations are not separable, the number of operators quadratically depends from N_S, thus making very consuming the implementation of HW designs with more than four orientations (4Or). A raw estimation of the area required from an eight-orientations (8Or) design with $f_0 = 0.2$, $\gamma = 1$, $\eta = 15$ and 7×7 and 9×9 scales returns more than 9 mm^2 in CMOS 90 nm and about 240-k LUTs in FPGA. In both cases, the dimensions are unacceptable for a usable implementation. However, in order to show that suitable results can be obtained from 2Or and 4Or, in Table II these are compared with results from 8Or in terms of peak signal-to-noise ratio (PSNR) and structural similarity (SSIM). The 2Or reduces the SSIM to 0.66, which could result insufficiently for some applications but highly desirable for resource-constrained devices, while 4Or presents an SSIM = 0.86 and a more than acceptable PSNR value for the large part of applications.

C. Gaussian Sharpness Values γ and η

The γ and η parameters are the standard deviations of the Gaussian envelope along x' and the Gaussian projection of the GFs along y', respectively. In order to avoid blurring problems and an excessive sensitivity of the filters to small variations into the images, their values must be related to the kernel dimensions N_S. Depending on the quality of the input set, these relations are generally determined by the subjective observation of the results. In the following, $\gamma = 1$ has been imposed to include at least one period of the periodic component in the Gaussian envelope, to ensure the presence of the odd function needed for edge detection [31]. In fact, the Gaussian shape has been adequately approximated by imposing the minimum filter dimensions to $6\gamma + 1$ [32], [33] giving rise to a filter having minimum dimensions 7×7. In our design, this filter has been combined to a 9×9 one to implement a multiscale architecture improving the accuracy and stability of the results. In turn, $\eta = 15$ has been imposed in order to simplify (2) in an HW-friendly fashion in the way that will be shown in Section V.

TABLE I

NUMBER OF ADDERS AND MULTIPLIERS AT DIFFERENT SCALE DIMENSIONS N_S AND NUMBER OF ORIENTATIONS N_O

Orientations	*Adders*	*Multipliers*
Two	$4(N_s-1)$	$2N_s$
Four	$2(N_s-1)(N_s+3)$	$2N_s(N_s+1)$
Eight	$2(N_s-1)(3N_s+5)$	$2N_s(3N_s+1)$

TABLE II

PSNR AND SSIM VALUES OBTAINED COMPARING THE 2Or AND 4Or RESULTS WITH THE 8Or

Image	*2Or*		*4Or*	
	PSNR [dB]	*SSIM*	*PSNR [dB]*	*SSIM*
airplane	21.90	0.72	25.16	0.87
baboon	20.11	0.58	25.47	0.86
boat	23.40	0.68	26.78	0.84
Lena	23.42	0.66	27.56	0.88
peppers	23.49	0.68	26.00	0.84
Average	**22.46**	**0.66**	**26.19**	**0.86**

D. Scales Dimensions N_s

Considering symmetric filters, namely, those having the same size along the two spatial dimensions, $N_x = N_y = N_s$, the value of N_s is a critical tradeoff between required resources and accuracy. Moreover, high N_s values will result in an excessive edge blurring in filtered images, while low N_s will result in a system too sensitive to noise. We have found that the use of two symmetric scales $N_S = 7$ and 9 represents a good tradeoff. It is worthwhile to note that regardless of the resource requirements, the eventual increment of the number of scales or their dimensions does not cause a substantial change in the topology of the proposed design. An example of the results returned by setting the above parameters is shown in the retinal image in Fig. 2 by varying the number of orientations.

IV. NUMERICAL REMARKS

Contrarily to most of the SW designs that use a floating point coding (FP32) according to IEEE-754 standard [34], our

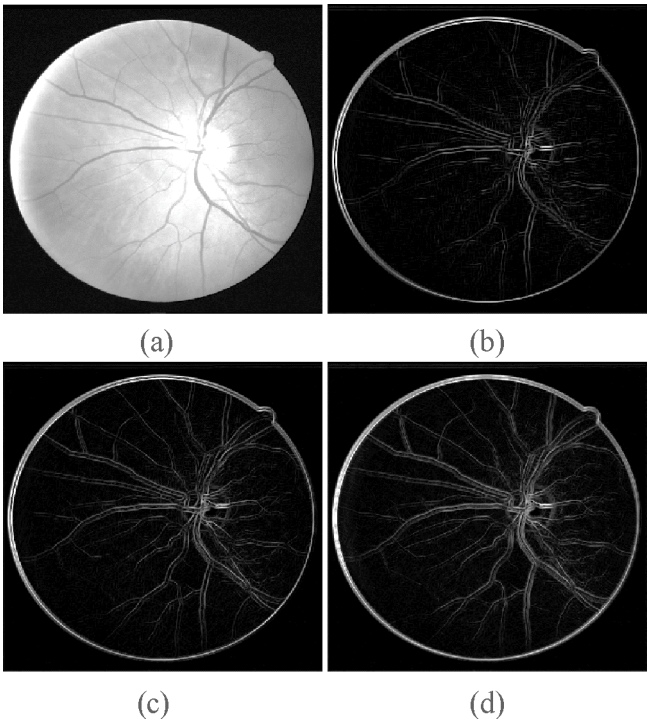


Fig. 2. (a) Original image taken from DRIVE database and results for $f_0 = 0.2$ cycle/pixels, $\gamma = 1$, $\eta = 15$ considering 7×7 GF and (b) two, (c) four, and (d) eight orientations.

design uses a fixed point (FI) format in order to reduce the amount of mapped physical resources, as will be quantified in Section V. In order to prevent the loss of accuracy arising from this choice, starting from the considerations in [30] for a 2-D Gaussian filter, a modeling of the worst case scenario has been developed, allowing to choose the minimum acceptable codelength for the final results. The maximum errors of the Gabor coefficients are reported in Table III as a function of their codelength. The first two bits represent the sign and the integer part, respectively, while the length of the fractional part varies. By imposing a 16-bits code (FI16), the error is lower than 3×10^{-5}, namely, lower of the least significant bit (LSB). To evaluate the error propagation up to the final result, the worst case for each orientation has been calculated by summing only the positive kernel coefficients multiplied by the maximum input value. Results are reported in Table IV. The shortest code of the final result for which the error is lower than the LSB is given by 12 bits, where 8 and 4 bits are for the integer and fractional part, respectively.

V. Proposed Designs

In this paper, two-orientations (2Or) and four-orientations (4Or) designs have been implemented, working on $\theta_{\text{Two}} = \{0, \pi/2\}$ and $\theta_{\text{Four}} = \{0, \pi/4, \pi/2, 3\pi/4\}$ orientation sets, respectively. A third design (4Or-shared) has been derived from the 4Or by implementing a resource-sharing architecture to obtain a better accuracy–area tradeoff.

The general architecture is schematized in Fig. 3, while the various units composing it are detailed in the following.

TABLE III
Maximum Error of the GF Coefficients by Varying the FI Codelength

Number of Bits	*Maximum Error*
14	1.04E-04
15	5.95E-05
16	**2.98E-05**
17	1.27E-05
18	7.58E-06

TABLE IV
Maximum Error of the Filtering Results for the Chosen Codelength of the Gabor Coefficients

Number of Bits	*Maximum Error*	*LSB*
10	1.05E-01	2.50E-01
11	4.95E-02	1.25E-01
12	**3.57E-02**	**6.25E-02**
13	3.57E-02	3.13E-02
14	2.00E-02	1.56E-02

A. Gabor Coefficients Memories

The coefficients of the GFs are stored in devoted memories. By observing that for the symmetry of (2) $G^{\theta}_{i,j} = G^{\theta+\pi/2}_{j,i}$ for $i, j \in I$ and $\theta \in [0, \pi/2]$, with $G^{\theta}_{i,j} = G(i, j, f_0, \theta)$, the dimensions of the memories storing the filter coefficients can be reduced. In particular, due to the choice of parameters, for $\theta = 0$ and $\theta = \pi/2$, all the rows and the columns, respectively, coincide. Therefore, in these cases only N_S coefficients need to be stored. In the case of $\theta = \pi/4$ and $\theta = 3\pi/4$, the symmetry develops along the minor and major diagonal of the kernel matrices, respectively, and only $(2N_S - 1)$ coefficients are needed. Considering the previous results and the use of the FI16 coding, two memories of 112 and 208 bits are needed when $N_S = 7$; for $N_S = 9$, the memory dimensions increase to 144 and 272 bits. Fig. 4 reports the memory size required for different scales by varying the number of orientations.

B. Memory Module

The Luma components of the incoming pixels are coded with 8 bits, acquired in raster scan order from the image source (e.g., an image sensor), and are then stored in a memory module (MM). As shown in Fig. 3, the MM is arranged to operate like a long first-in–first-out, having overall dimensions $W \times N_S^{\max}$, with W the width of the input image and $N_S^{\max}$the dimension of the greatest kernel, 9 in our case. Therefore, the structure behaves like a serial-input parallel-output (SIPO) memory. In order to avoid the problems related to the optimal placement of a large amount of registers and to the consequent congestion of the post-routed interconnections, dual-port SRAM modules, together with an address generator, have been implemented to emulate the shifting behavior of the SIPO [30], [35]. In particular, $N_S^{\max}$ dual-port SRAMs, one for each row, with dimensions $1 \times (W - N_S^{\max})$, have been instantiated for the purpose. The SIPO has been terminated with a bank of $N_S^{\max} \times N_S^{\max}$ registers, organized like in Fig. 3, which are updated with a new column of $N_S^{\max}$ data from the SRAMs at each clock cycle. Since the data already stored in the bank shift of a column at each cycle, after an

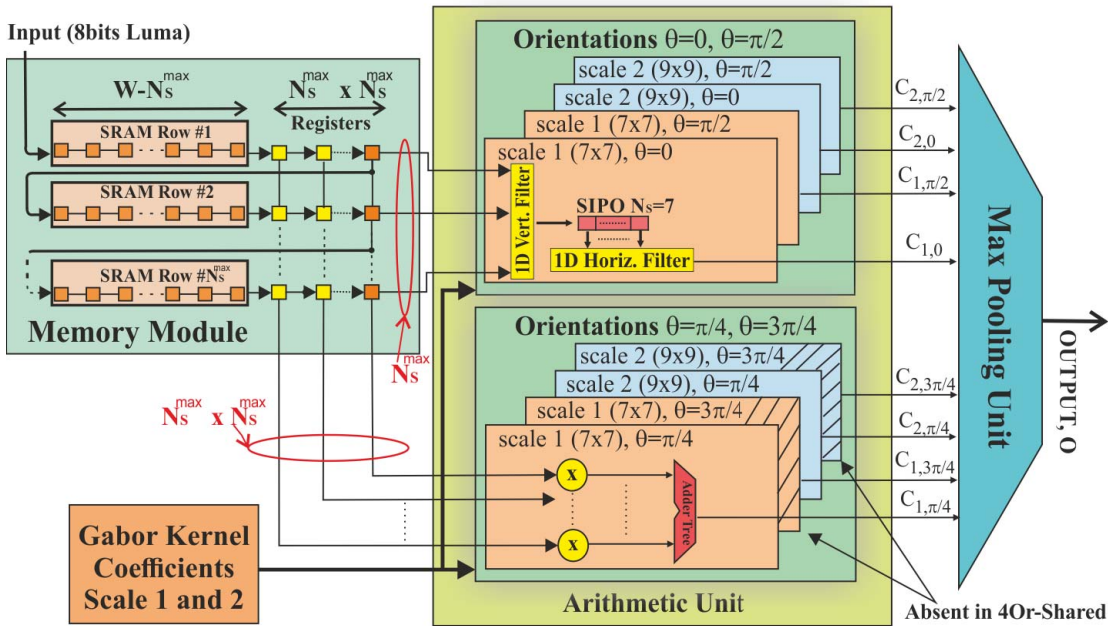


Fig. 3. General block scheme of the proposed designs.

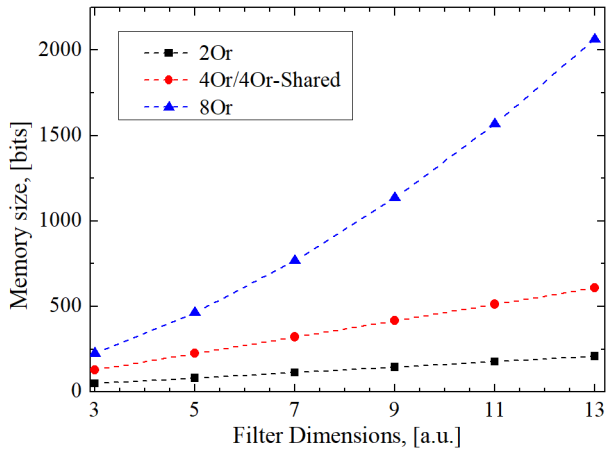


Fig. 4. Memory size at different filter dimensions.

initial latency of $N_S^{\max}$ clock cycles, the bank stores an entire $N_S^{\max} \times N_S^{\max}$ tile of the input image. After the initial latency, at each clock cycle, the rightmost column from the bank is read from the separable modules of the filter ($\theta = 0, \pi/2$), while the nonseparable modules ($\theta = \pi/4, 3\pi/4$) read all the tiles in parallel [32], [33].

C. Arithmetic Unit

The arithmetic unit (AU) computes the partial results taking the data coming from the Gabor coefficients memories and the MM, developing the filtering operations for each orientation as schematized in Fig. 5. One of the problems in dealing with GF is that the number of adders and multipliers quadratically increases with the number of scales and their dimensions as shown in Table I. The increase is linear when the kernel is separable. By exploiting the separability of (2) along the directions θ_{Two} as

$$\begin{cases} G_0 = G(x, y, f_0, 0) = \frac{f_0^2}{\pi\gamma\eta} \sin(2\pi f_0 x) e^{-\frac{f_0^2}{\gamma^2}x^2} \times e^{-\frac{f_0^2}{\eta^2}y^2} \\ \quad = G_0^H \times G_0^V \\ G_{\pi/2} = G\left(x, y, f_0, \frac{\pi}{2}\right) = \frac{f_0^2}{\pi\gamma\eta} \sin(2\pi f_0 y) e^{-\frac{f_0^2}{\gamma^2}y^2} \times e^{-\frac{f_0^2}{\eta^2}x^2} \\ \quad = G_{\pi/2}^V \times G_{\pi/2}^H \end{cases} \tag{3}$$

it is possible to reduce the complexity of the computation along these directions from $O(N_s^2)$ to $O(N_s)$, according to the scheme of the separable orientations in Fig. 5. Furthermore, the choice $\eta = 15$ simplifies (3) as

$$\begin{cases} G_0 = G_0^V \times G_0^H = I^T \times G_0^H \\ G_{\pi/2} = G_{\pi/2}^V \times G_{\pi/2}^H = G_{\pi/2}^V \times I \end{cases} \tag{4}$$

where I is the identical vector. Therefore, one of the 1-D filters of the orientations $\theta = 0$ and $\theta = \pi/2$ can be implemented with only FI16 adders. Notwithstanding, Fig. 6 shows that because of the nonseparable orientations, the number of adders and multipliers continues to quadratically increase with the filter dimensions and the number of orientations. Aiming to mitigate these problems, the 4Or-shared design presents a reduced AU, obtained by sharing the same structure to calculate both the $\pi/4$ and the $3\pi/4$ orientations. In this case, the same module of the AU is used to filter both the $\theta = \pi/4$ and $\theta = 3\pi/4$ orientations. This feature is implemented by introducing a routing network that selects the correct Gabor coefficients to be sent to the structure at any given clock cycle. This implementation reduces the number of adders and multipliers by 41% and 44.5%, respectively, with respect to the 4Or design, as reported in Table V. The cost to pay in this case is in the reduction of throughput, because the 4Or-shared design needs two clock cycles to filter a single pixel along all the orientations, and in a decrement of the timing because of the major congestion for the signal routing.

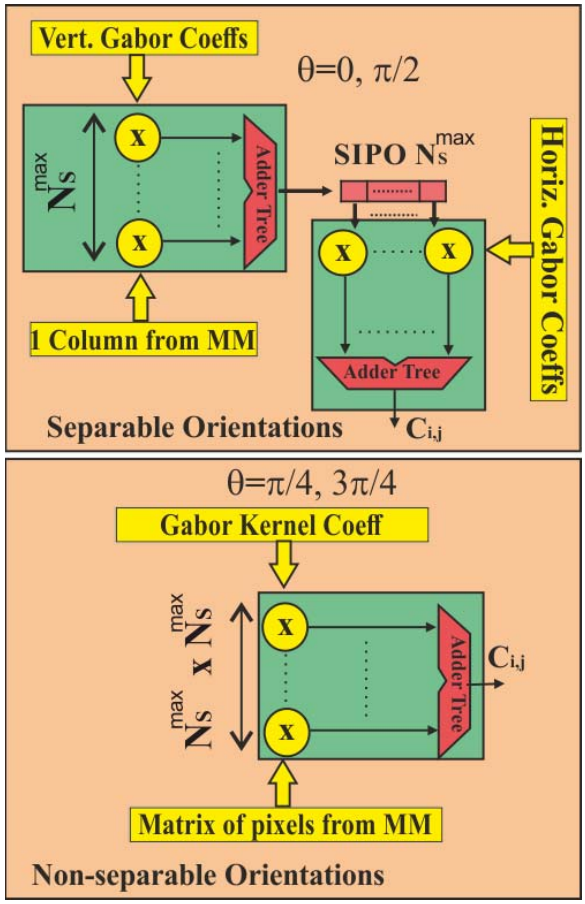


Fig. 5. Schemes of the separable and nonseparable filter module of the AU.

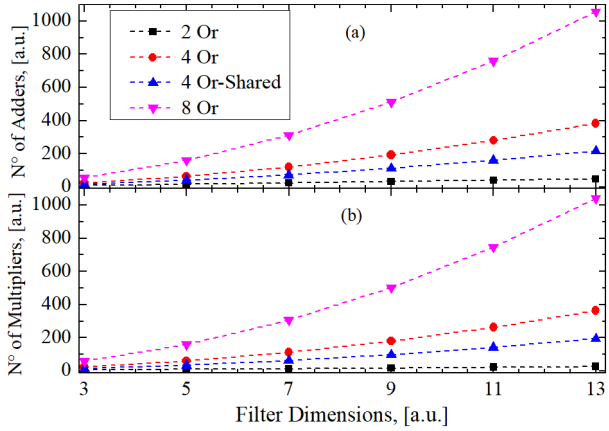


Fig. 6. Number of (a) adders and (b) multipliers at different filter dimensions.

TABLE V
TOTAL NUMBER OF ADDERS AND MULTIPLIERS USED IN THE CONSIDERED DESIGNS WITH TWO SCALES

Orientations	*Adders*	*Multipliers*
2 Orientations	56	32
4 Orientations	312	292
4 Orientations Shared	184	162

The total number of adders and multipliers to be implemented for the proposed designs in Table V makes evident the increase in complexity related to the addition of nonseparable filters.

D. Control Unit

The control unit (CU) is a finite-state machine (FSM) managing the correct transferring of the data to and from the AU, through several control signals. The FSM also takes into account the case of processing near the image corners or edges in order to obtain a correct edge detection also in these regions. This is accomplished by creating different filtering masks for the various cases in which the pixels exceeding the frame are not taken into account in the computation. Moreover, the CU is capable of synchronizing the scales in order to provide consistent data to the Max Pooling. This task is even more important in the case of the 4Or-shared architecture, in which the CU must also synchronize the internal subunits of the AU dedicated to the $\theta = \pi/4$ and $\theta = 3\pi/4$ orientations, in order to collect all the partial results, before sending them to the max-pooling unit (MPU).

E. Max-Pooling Unit

The MPU compares the results coming from the AU, computed for the various orientations and scales, and selects the greatest value among them, $O = \text{Max}\{C_{1,0}, C_{1,\pi/2}, C_{2,0}, C_{2,\pi/2}, C_{1,\pi/4}, C_{1,3\pi/4}, C_{2,\pi/4}, C_{2,3\pi/4}\}$ as a feature candidate to compose the result image of the edge feature candidates.

VI. SYNTHESIS AND RESULTS

The designs have been targeted to a Xilinx Virtex 7 XC7V2000tflg1925-1, as a part of the proFPGA DUO ASIC prototyping board and to TSMC CMOS 90-nm std_cells using Cadence toolchain. Synthesis results have been reported in Tables VI and VII, where CMOS has been constrained in the typical process-voltage-temperature corner, namely, standard process, 1.2-V bias, and temperature of 27 °C. The use of the FI coding over the FP32 is futher justified by observing that a single FP32 multiplier from the Xilinx IP library would require 681 LUTs compared to the 126 LUTs used for the FI implementation, while an FP32 adder would require 239 LUTs compared to the 77 LUTs for the FI implementation. Therefore, an FP32 implementation of the AU for the 4Or architecture would require 273420 LUTs compared to the 59453 LUTs of the FI, namely, the FP32 implementation should occupy about the 22% of the Xilinx Virtex 7 XC7V2000tflg1925-1, which is very unacceptable for our purpose, considering that the FPGA is ones of the largest one currently on the market. The std_cell implementation of the same design would rawly require an area of approximately 11 mm^2, which is clearly an excessive value.

Results from Tables VI and VII show that the 2Or architecture achieves the best APD performances. However, it is not recommended when accuracy is the primary concern of the application. In turn, the 4Or-shared provides the same accuracy of the 4Or, while showing lower area requirement.

TABLE VI
SYNTHESIS OF THE PROPOSED DESIGNS ON FPGA PLATFORM

	FPGA						
	With DSPs				***Without DSPs***		
	2Or	*4Or*	*4Or-Shared*	*Cesur [26, 27]*	*2Or*	*4Or*	*4Or-Shared*
N° of orient,, scales	2, 2	4, 2	4, 2	1,1	2, 2	4, 2	4, 2
Kernel dimensions	7x7, 9x9	7x7, 9x9	7x7, 9x9	3x3	7x7, 9x9	7x7, 9x9	7x7, 9x9
Target platform	Virtex 7	Virtex 7	Virtex 7	Stratix 4	Virtex 7	Virtex 7	Virtex 7
LUTs	4022	27042	22281	14025	8233	59930	41265
FFs	2864	16220	11833	20321	3440	21812	14912
DSPs	32	260	146	202	--	--	--
Path Delay[ns]	5.6	6.4	10.0	6.73 (2.24)***	5.6	6.4	10.0
Power [W]*	0.775	1.629	1.748	--	0.821	1.898	2.327
*fps***	86	75	48	60 (180)***	86	75	48

*Normalized at 100MHz **Full-HD frames ***max. theoretical

TABLE VII
SYNTHESIS OF THE PROPOSED DESIGNS IN STD_CELLS

	Std_cells		
	2Or	***4Or***	***4Or-Shared***
Platform	90nm	90nm	90nm
Area [μm^2]	1002929	2414128	1801557
Path Delay[ns]	2.86	3.37	6.83
Power [mW]*	14.013	76.453	27.135
*fps***	168	143	70

*Normalized at 100MHz ** Full-HD (1920x1080 pixels) frames

TABLE VIII
POWER RESULTS FOR THE PROPOSED ARCHITECTURES NORMALIZED AT A FREQUENCY OF 62.2 MHz

	FPGA					
	2Or		***4Or***		***4Or-Shared***	
DSPs	w/o	w	w/o	w	w/o	w
Power [W]*	0.762	0.739	1.446	1.244	1.728	1.347
	Std_cell					
	2Or		***4Or***		***4Or-Shared***	
Power [mW]*	9.742		52.150		19.408	

*Normalized at 62.2MHz

Its main drawback is represented by the higher delay when compared to the other solutions. However, considering that it exhibits a delay of 6.83 ns in std_cells and 10 ns in FPGA, real-time performances are achieved up to frame resolutions of 2209×2209 pixels and 1825×1825 pixels for the ASIC and FPGA, respectively, all compatible with full-HD standard. Therefore, considering the APD product and the accuracy of the results, the 4Or-shared solution appears as the optimal choice, if there are no strict requirements in terms of power budget and area consumption.

In fact, the 4Or-shared obtains a better edge detection compared to the 2Or at the cost of a 79.6% area overhead and 93.6% increase in power consumption for an ASIC implementation. At the same time, the 4Or-shared provides the same accuracy of the 4Or implementation, while reducing the area occupation of 25.4% and the power consumption of 64.5%. Although comparisons with [26] and [27] are "unfair" for their small kernel dimensions (3×3) and the use of only one orientation and one only scale, they have been reported in Table VI since they are the most recent HW designs having in resource saving one of the primary objectives. Their maximum theoretical throughput of 180 f/s at 445.5 MHz on FPGA is justified by the use of DSPs that do not find a direct counterpart in CMOS. In turn, although our 2Or proposal reaches a lower throughput of 86 f/s, it maps 71.3% and 85.9% less LUTs and FFs, respectively, with DSPs, and 41.3% and 83% less LUTs and FFs without DSPs. Therefore, our proposal can be easily targeted to CMOS 90-nm std_cells, where it is estimated of occupying a reasonable area of 1 mm^2 and consumes 14 mW.

Since the obtained fps could be reduced without compromising the real-time operation for all the proposed designs, a further study on power consumption has been conducted, decreasing the operating frequency to 62.2 MHz, in order to obtain a f/s $= 30$ for 1920×1080 pixels frames. The obtained results are reported in Table VIII, where it is possible to notice the decrease in the power consumption, mainly due to the reduction in the dynamic power consumption, related to the lower operating frequency.For comparison with the SW counterpart, it is worthwhile to report that the MATLAB implementation of our 4Or design requires 2.98 s to process a full-HD image when running on a CPU Intel I7 3537U at 2.5 GHz.

VII. CONCLUSION

This paper proposes the design of new ASIPs to implement the GF for medical imaging applications relevant for pre-processing applications in portable and resource-constrained devices for CAD. Thanks to HW-friendly implementations, state-of-the-art performances are achieved both on FPGA and std_cell implementations. A future improvement of the design will involve Distributed Arithmetic techniques to simplify the arithmetic circuitry [36], [37] and explore the possibility of a further increase of the number of orientations and scales.

REFERENCES

[1] C.-Y. Lu *et al.*, "Vessel enhancement of low quality fundus image using mathematical morphology and combination of Gabor and matched filter," in *Proc. Int. Conf. Wavelet Anal. Pattern Recognit. (ICWAPR)*, Jul. 2016, pp. 168–173.

[2] G. Humpire-Mamani, A. J. M. Traina, and C. Traina, "k-Gabor: A new feature extraction method for medical images providing internal analysis," in *Proc. 25th Int. Symp. Comput.-Based Med. Syst. (CBMS)*, Jun. 2012, pp. 1–6.

[3] B. Ergen, A. Çinar, and G. Aydin, "Gabor wavelet and unsupervised fuzzy C-means clustering for edge detection of medical images," in *Proc. Int. Symp. Innov. Intell. Syst. Appl. (INISTA)*, Jul. 2012, pp. 1–4.

[4] M. N. S. Zainudin, M. M. Said, and M. M. Ismail, "Feature extraction on medical image using 2D Gabor filter," *J. Appl. Mech. Mater.*, vol. 52, pp. 2128–2132, Mar. 2011.

[5] S. Lahmiri and M. Boukadoum, "Hybrid discrete wavelet transform and Gabor filter banks processing for features extraction from biomedical images," *J. Med. Eng.*, vol. 2013, Mar. 2013, Art. no. 104684.

[6] A. Olowoyeye, M. Tuceryan, and S. Fang, "Medical volume segmentation using bank of Gabor filters," in *Proc. ACM Symp. Appl. Comput.*, Jan. 2009, pp. 826–829.

[7] S. Vishukumar, K. Patel, and P. Shrivastava, "Implementation of medical image enhancement technique using Gabor filter," *Int. J. Current Eng. Technol.*, vol. 2, no. 2, pp. 251–255, Jun. 2012.

[8] S. Veni and K. A. Narayanankutty, "Image enhancement of medical images using Gabor filter bank on hexagonal sampled grids," *World Acad. Sci., Eng. Technol.*, vol. 65, no. 41, pp. 816–821, May 2010.

[9] C.-I. Chen, "Fusion of PET and MR brain images based on IHS and log-Gabor transforms," *IEEE Sensors J.*, vol. 17, no. 21, pp. 6995–7010, Nov. 2017.

[10] A. Pourtaherian *et al.*, "Medical instrument detection in 3-dimensional ultrasound data volumes," *IEEE Trans. Med. Imag.*, vol. 36, no. 8, pp. 1664–1675, Aug. 2017.

[11] J.-K. Kamarainen, V. Kyrki, and H. Kälviäinen, "Robustness of Gabor feature parameter selection," in *Proc. IAPR Workshop Mach. Vis. Appl.*, Dec. 2002, pp. 132–135.

[12] W. Jiang, K. M. Lam, and T. Z. Shen, "Efficient edge detection using simplified Gabor wavelets," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 39, no. 4, pp. 1036–1047, Aug. 2009.

[13] W.-C. Zhang, F.-P. Wang, L. Zhu, and Z.-F. Zhou, "Corner detection using Gabor filters," *Image Process.*, vol. 8, no. 11, pp. 639–646, 2014.

[14] H. Hu, "Enhanced Gabor feature based classification using a regularized locally tensor discriminant model for multiview gait recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 7, pp. 1274–1286, Jul. 2013.

[15] J. G. Daugman, "Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters," *J. Opt. Soc. Amer. A, Opt. Image Sci.*, vol. 2, no. 7, pp. 1160–1169, 1985.

[16] D. Gabor, "Theory of communication. Part 1: The analysis of information," *J. Inst. Elect. Eng.*, vol. 93, no. 26, pp. 429–441, Jul. 1946.

[17] J. Chen, Y. Sato, and S. Tamura, "Orientation space filtering for multiple orientation line segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 5, pp. 417–429, May 2000.

[18] L. F. Albanese and G.-D. Licciardo, "High speed CAVLC encoder suitable for field programmable platforms," in *Proc. Int. Conf. Signals Electron. Circuits*, Sep. 2010, pp. 327–330.

[19] G. D. Licciardo and L. F. Albanese, "Design of a context-adaptive variable length encoder for real-time video compression on reconfigurable platforms," *Image Process.*, vol. 6, no. 4, pp. 301–308, Jun. 2012.

[20] G. D. Licciardo and M. Costagliola, "An H. 264 encoder for real time video processing designed for spear customizable system-on-chip family," in *Proc. IEEE Int. Conf. Signal Process. Commun.*, Nov. 2007, pp. 824–827.

[21] L. F. Albanese and G. D. Licciardo, "An area reduced design of the context-adaptive variable-length encoder suitable for embedded systems," in *Proc. 5th Int. Symp. I/V Commun. Mobile Netw. (ISVC)*, Sep. 2010, pp. 1–4.

[22] G. Desoli *et al.*, "A 2.9 TOPS/W deep convolutional neural network SoC in FD-SOI 28 nm for intelligent embedded systems," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, San Francisco, CA, USA, Feb. 2017, pp. 238–239.

[23] C. Shi *et al.*, "Shape sensing techniques for continuum robots in minimally invasive surgery: A survey," *IEEE Trans. Biomed. Eng.*, vol. 64, no. 8, pp. 1665–1678, Aug. 2017.

[24] K. Chen, H.-S. Lee, and C. G. Sodini, "A column-row-parallel ASIC architecture for 3D wearable/portable medical ultrasonic imaging," in *Symp. VLSI Circuits Dig. Tech. Papers*, Jun. 2014, pp. 1–2.

[25] A. Anand, I. Moon, and B. Javidi, "Automated disease identification with 3-D optical imaging: A medical diagnostic tool," *Proc. IEEE*, vol. 105, no. 5, pp. 924–946, May 2017.

[26] E. Cesur, N. Yildiz, and V. Tavsanoglu, "An improved FPGA implementation of CNN Gabor-type filters," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Rio de Janeiro, Brazil, May 2011, pp. 881–884.

[27] E. Cesur, N. Yildiz, and V. Tavsanoglu, "On an improved FPGA implementation of CNN-based Gabor-type filters," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 59, no. 11, pp. 815–819, Nov. 2012.

[28] *Virtex-7 T and XT FPGAs Data Sheet: DC and AC Switching Characteristics*, Xilinx, San Jose, CA, USA, Jun. 2015.

[29] F. A. Pellegrino, W. Vanzella, and V. Torre, "Edge detection revisited," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 34, no. 3, pp. 1500–1518, Jun. 2004.

[30] G. D. Licciardo, T. Boesch, D. Pau, and L. Di Benedetto, "Frame buffer-less stream processor for accurate real-time interest point detection," *Integr., VLSI J.*, vol. 54, pp. 10–23, Jun. 2016.

[31] R. Mehrotra, K. R. Namuduri, and N. Ranganathan, "Gabor filter-based edge detection," *Pattern Recognit.*, vol. 25, no. 12, pp. 1479–1494, Dec. 1992.

[32] G. D. Licciardo, A. D'Arienzo, and A. Rubino, "Stream processor for real-time inverse tone mapping of full-HD images," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 23, no. 11, pp. 2531–2539, Nov. 2015.

[33] G. D. Licciardo, C. Cappetta, L. Di Benedetto, A. Rubino, and R. Liguori, "Multiplier-less stream processor for 2D filtering in visual search applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 1, pp. 267–272, Jan. 2018.

[34] *IEEE Standard for Binary Floating-Point Arithmetic*, ANSI/IEEE Standard 754, 1985.

[35] G. D. Licciardo, C. Cappetta, and L. Di Benedetto, "Application specific image processor for the extension of the dynamic range of images with multiple resolutions," in *Proc. IEEE Int. Conf. Electron., Circuits Syst. (ICECS)*, Monte Carlo, Monaco, Dec. 2016, pp. 700–703.

[36] G. D. Licciardo, C. Cappetta, L. Di Benedetto, and M. Vigliar, "Weighted partitioning for fast multiplierless multiple-constant convolution circuit," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 64, no. 1, pp. 66–70, Jan. 2017.

[37] S. Venkatachalam and S.-B. Ko, "Design of power and area efficient approximate multipliers," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 25, no. 5, pp. 1782–1786, May 2017.

Gian Domenico Licciardo (S'03–M'06–SM'17) was born in Naples, Italy, in 1974. He received the electronic engineering degree from the University of Naples Federico II, Naples, in 2002, and the Ph.D. degree in information engineering from the University of Salerno, Fisciano, Italy, in 2006.

Since 2006, he has been the Counselor of the IEEE Student Branch, University of Salerno, where he has been an Assistant Professor of electronic with the Department of Industrial Engineering since 2007. His current research interests include the modeling, simulation, and characterization of electron devices and the design of very large scale integration systems.

Dr. Licciardo served as a Treasurer for the Electron Device Central and Southern Italy Chapter from 2006 to 2012.

Carmine Cappetta (S'15) was born in Battipaglia, Italy, in 1989. He received the B.S. and M.S. degrees in electronic engineering from the University of Salerno, Fisciano, Italy, in 2012 and 2014, respectively, where he is currently pursuing the Ph.D. degree with the Department of Industrial Engineering.

His current research interests include the design of very large-scale integration systems for high-speed image and video processing.

Luigi Di Benedetto (S'06–M'14) was born in Salerno, Italy, in 1984. He received the B.Sc. and M.Sc. degrees (*cum laude*) in electronic engineering and the Ph.D. degree in solid-state electronics from the University of Salerno, Fisciano, Italy, in 2006, 2009, and 2013, respectively.

Since 2013, he has been a Research Fellow. In 2013, he joined the Fraunhofer IISB, Erlangen, Germany, and the Friedrich-Alexander University, Erlangen-Nürnberg, Germany, as a Visiting Scientist, investigating on the blocking behavior of 4H-SiC high-voltage bipolar devices. His current research interests include the modeling, simulation, and development of high-power electronic devices based on wide-bandgap semiconductor.