

Embedded multi-spectral image processing for real-time medical application[☆]



Chao Li*, Souleymane Balla-Arabé, Fan Yang

LE2I CNRS 6306 Laboratory, University of Burgundy, France

ARTICLE INFO

Article history:

Received 8 July 2015

Revised 9 November 2015

Accepted 9 December 2015

Available online 23 December 2015

Keywords:

Multi-spectral image processing

Light–Tissue Interaction

Genetic Algorithm

Embedded System

FPGA

High-Level Synthesis

ABSTRACT

The newly introduced Kubelka–Munk Genetic Algorithm (KMGA) is a promising technique for the assessment of skin lesions from multi-spectral images. Using five skin parameter maps such as concentration or epidermis/dermis thickness, this method combines the Kubelka–Munk Light–Tissue interaction model and Genetic Algorithm optimization process to produce a quantitative measure of cutaneous tissue. Up to the present, variant improved KMGA implementations have been successfully realized using the recent parallel computing techniques. However, all these achievements are based on the multi-core CPUs. This results in a quite high cost and low practicability for the hardware equipment of the clinical system. Fortunately, Embedded Systems (ES) applications have made great progress in recent years, and many highly effective image processing devices, such as DSPs (Digital Signal Processor) and FPGAs (Field Programmable Gate Array), have been made available to engineers at a very convenient price. Nevertheless, today's embedded devices have as well the advantages of high speed, high embedability, low power consumption, more flexibility, etc. Thus, we focus our researches on the embedded KMGA application development. In this paper, we realize the CPU-to-FPGA transplantation of KMGA within a special High-Level Synthesis (HLS) SW/HW Co-design framework. Moreover, several optimizations are made on the algorithm and source code to improve the performances of the final implementation. Compared with CPUs, intensive experiments demonstrate that the proposed approaches can effectively improve the performances of KMGA method both in terms of efficiency and accuracy.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

Medical imaging is one of the major research subjects in Computer-Aided Diagnosis (CAD). With computer-aided medical imaging, doctors use the computerized analysis results as a “second opinion” to make the final decision. This technique can improve the diagnosis by helping for the diagnostic itself or quantifying the evaluation results, and be used for monitoring the efficiency of a treatment over time as well. For example, Medical ultrasound is widely used for non-destructive diagnosis of internal body structure lesions or guiding the treatment process, while Molecular Imaging is developed to explore the changing of cells and molecular level during the disease process.

Historically, well trained dermatologists analyze the skin color and interpret the clinical pathologies depending on their knowledge and experience, which often results in the mistakes due to

the subjective judgment. Recently, in order to make the diagnosis conclusions objective, computer assisted methods for cutaneous lesions assessment increasingly attracts the medical researchers. More precisely, some image processing systems are used to minimize the usages of the naked eyes and quantify the lesions zone's optimal properties.

Using the knowledge of the skin absorption and scattering properties, a novel Light–Tissue Interaction model based multi-spectral skin lesion assessment method, Kubelka–Munk Genetic Algorithm (KMGA), is proposed by Jolivot et al. [1]. This method combines the KM model [2] with Genetic Algorithm (GA) for the optimization process. It can analyze both of the most important light absorbers (blood and melanin) in the skin according to the multi-spectral images which is acquired only by a hand-held multi-spectral camera. However, KMGA is a quite resources costly algorithm. Its central unit for the data processing is the high performance multi-core CPUs in personal computers (see Fig. 3.3 in [3]). This results in a high cost for the hardware equipment and seriously narrows its advantages in terms of portability. Therefore, finding a lighter, cheaper and powerful alternative of CPUs for KMGA becomes a new challenge.

[☆] Thanks to China Scholarship Council for funding.

* Corresponding author. Tel.: +33 658892280.

E-mail address: chao.li.1986@ieee.org, lichao8601@hotmail.com (C. Li).

Our work focus on the performance improvement of KMGAs skin lesion assessment system by using high performance computing technologies. In recent years, Embedded Systems (ES) have made great progress, and many highly effective Field Programmable Gate Array (FPGA) devices have been made available to engineers at a very convenient price. These achievements offer nice opportunities to obtain more performance improvements from a complex design [4–12]. For example, Colodro-Conde et al. [8] propose a FPGA architecture of area-based algorithms for calculating distance in stereoscopic vision systems, Sidiropoulos et al. [9] introduce a novel 3-D FPGA architecture for efficient implementation both of compute-bound and I/O-bound applications and Toledo-Moreo et al. [10] present a hardware architecture for the FPGA-based implementation of 2-D convolution with medium–large kernels. Furthermore, Zuo et al. [13] and Kestur et al. [14] point out that in their designs FPGAs performance much better than CPUs or GPUs in terms of power-efficient, and another comparative study made by González et al. [15] further indicates that the cost of FPGAs is significantly lower than the other computing platforms. Thus, we fix the goal of the work introduced in this paper on the FPGA implementation of KMGAs.

Within the conventional development framework, the development languages of FPGA, i.e. VHDL or Verilog, usually have a low abstract level, which allows the hardware configurations in RTL (Register-Transfer Level). Therefore, the complex algorithms are difficult to be specified within such languages. Recently, Cong et al. [16] and Liang et al. [17] introduce a novel High-Level Synthesis (HLS) procedure that can automatically synthesize the specification of algorithm from C like languages into RTL. Nevertheless, its C-to-RTL synthesis process can be configured by using directives for implementation optimizations. We therefore base our work on a HLS based SW/HW Co-design framework.

In this paper, we successfully realize the FPGA implementation of a High-Convergence-Ratio KMGAs (HCR-KMGAs) skin lesion assessment method improved from the prototype of KMGAs. During the development, several optimizations are made in order to improve the performances of the generated RTL implementation, including optical function rewriting, function optimizer improving and memory optimizing. The proposed implementations is evaluated by comparing with its CPU implementations optimized by parallel computing techniques. Intensive experiments demonstrate that our approaches can effectively accelerate the KMGAs skin lesion assessment system, while improving its accuracy as well.

The remainder of this paper is organized as follows: Section 2 describes the fundamental principles of the KMGAs method and its algorithm-level improvements. Section 3 presents the development process of HCR-KMGAs and its hardware level optimizations. Section 4 analyzes the experimental results and evaluates our design's performances. Finally, a conclusion is given in Section 5.

2. Algorithm description

In order to retrieve the different skin physical or biological properties, several skin models have been developed [18–20]. Kubelka–Munk Genetic Algorithm is one of the latest Light–Tissue Interaction skin lesions assessment approaches. It retrieves the interested skin biological properties by inverting the KM model with the GA procedure. Firstly, the reflectance spectrum of the lesions' zone, defined as a set of total reflectance values with different wavelengths, is measured with an acquisition system. Meanwhile, a population composed of numbers of candidate solutions (called individuals) is initialized as the search space of the selection procedure. Each individual carries the information for the selection procedure, including the simulated optical properties (reflectance spectrum), the biological properties and the fitness value. In the KMGAs prototype, the reflectance spectrums are performed

according to the KM model, while the biological properties are randomly generated within the reasonable bounds. The fitness value refers to the spectrum similarity between the simulated spectrum and the measured spectrum. Then, the population is repeatedly selected through the selection process until a predefined number of iterations. Finally, the best candidate is selected.

KMGAs could effectively retrieve the skin parameter maps via a selection process, however, this task is running-costly [18] even for a powerful processor. Thus, we propose a novel High-Convergence-Rate KMGAs (HCR-KMGAs) method in this section. Comparing with its prototype implemented by Jolivot et al. [3], our implementation can make more acceleration gains according to the following three approaches:

- HCR-KMGAs re-specifies the KM function in order to reduce the redundant operations down to minimum.
- A Predictive Function Optimization Algorithm (PFOA) is designed to accelerate the convergence of function optimization process.
- HCR-KMGAs' individuals' parameters are optimized depending on the data dependency, some unnecessary data are removed in order to save memory space.
- Multiple different termination conditions are performed in HCR-KMGAs in order to avoid the redundant iterations.

2.1. Kubelka–Munk model

KMGAs-based skin lesion assessment system treats the cutaneous system as an epidermis and dermis based 2-layers KM model with five principal parameters that affect the light's reflectance and transmittance: melanin concentration, epidermis thickness, blood concentration, blood oxygen saturation and dermis thickness. This algorithm consists mainly in population initialization, generation, and evolution. Experimental results show that the population initialization and generation takes up to 96% of the total execution time, population evolution takes 3% and other operations only 1%. The optical model of KM is the key technique used during the time consuming process of population initialization and generation. Thus, we use a reduced KM function previously developed for running accelerating [21].

In KM function, the total light reflectance R_{tot} and transmittance T_{tot} are expressed as:

$$R_{tot} = R_{1,2} = R_1 + \frac{T_1^2 R_2}{1 - R_1 R_2} \quad (1)$$

$$T_{tot} = T_{1,2} = \frac{T_1 T_2}{1 - R_1 R_2} \quad (2)$$

The reflectance R_n and transmittance T_n for a single layer n can be expressed as a function of the thickness of the layer d_n , the absorption coefficient $\mu_{a,n}$ and the scattering coefficient $\mu_{s,n}$. In order to simplify the computation, KM function are re-specified as follows:

$$R_n = \frac{\mu_{s,n} \times (E - 1)}{(\mu_{a+s} + K_n) \times E - (\mu_{a+s} - K_n)} \quad (3)$$

$$T_n = \frac{2K_n \epsilon}{(\mu_{a+s} + K_n) \times E - (\mu_{a+s} - K_n)} \quad (4)$$

where

$$K_n = \sqrt{\mu_{a,n}(\mu_{a,n} + 2\mu_{s,n})} \quad (5)$$

$$\mu_{a+s} = \mu_{a,n} + \mu_{s,n}$$

$$E = \epsilon^2 = e^{2K_n d_n}$$

The optical absorption and scattering coefficients in the epidermis and dermis layers, $\mu_{a,epidermis}$, $\mu_{a,dermis}$, $\mu_{s,epidermis}$ and $\mu_{s,dermis}$,

Table 1

Equation symbol definitions for Eq. (6)–(8) (top) and the five interested skin parameters (bottom).

Symbol	Definition
$\mu_{a,mel}$	Melanin absorption coefficient
$\mu_{a,baseline}$	Baseline absorption coefficient
$\mu_{a,oxy}$	Oxy-haemoglobin absorption coefficient
$\mu_{a,deoxy}$	Deoxy-haemoglobin absorption coefficient
$\mu_{s,Mie}$	Mie scattering coefficient
$\mu_{s,Rayleigh}$	Rayleigh scattering coefficient
f_{mel}	Melanin concentration
D_{epi}	Epidermis thickness
f_{blood}	Volume blood fraction
C_{oxy}	Oxygen saturation
D_{derm}	Dermis thickness

are computed as follows:

$$\mu_{a,epidermis} = f_{mel}\mu_{a,mel} + (1 - f_{mel})\mu_{a,baseline} \quad (6)$$

$$\mu_{a,dermis} = f_{blood}(C_{oxy}\mu_{a,oxy}) + f_{blood}(1 - C_{oxy})\mu_{a,deoxy} + (1 - f_{blood})\mu_{a,baseline} \quad (7)$$

$$\mu_{s,epidermis} = \mu_{s,dermis} = \mu_{s,Mie} + \mu_{s,Rayleigh} \quad (8)$$

where

$$\begin{aligned} \mu_{a,mel} &= 6.6 \times 10^{11} \lambda^{-3.33} \\ \mu_{a,baseline} &= 0.244 + 85.3 \times e^{-(\lambda-164)/66.2} \\ \mu_{a,oxy} &= \ln 10 \times HbO_2(\lambda) \times G/M \\ \mu_{a,deoxy} &= \ln 10 \times Hb(\lambda) \times G/M \\ \mu_{s,Mie} &= 2 \times 10^5 \times \lambda^{-1.5} \\ \mu_{s,Rayleigh} &= 2 \times 10^{12} \times \lambda^{-4} \end{aligned} \quad (9)$$

The definitions of the symbols in Eqs. (6)–(8) and the five interested parameters are shown in Table 1. In Eq. (9), λ is the light's wavelength, HbO_2 and Hb are the oxy-haemoglobin and deoxy-haemoglobin content in cm^{-1} , G is the haemoglobin's weight in gram per liter and M is the gram molecular weight of haemoglobin.

Reducing KM function is a simple but effective approach to accelerate the execution of design. The improved algorithm required less computational operations as the redundant instructions are avoided by arithmetical reducing. For example, the power operations appear 13 times in the conventional prototype but only 3 times in the reduced formulas.

2.2. Function optimizer

According to Eqs. (1)–(9), the total reflectance of the incident light can be expressed as a function of the five interested skin parameters with a fixed wavelength:

$$R_{tot} = f_{KM}(f_{mel}, D_{epi}, f_{blood}, C_{oxy}, D_{dermis}) \quad (10)$$

It is obvious that Eq. (10) is a complex non-linear function with five arguments which is impossible to inverse by mathematical methods. KMGA optimizes this function according to a standard genetic algorithm. This optimization process is a search heuristic that mimics the process of natural selection. It generates solutions to optimization problems using techniques inspired by natural evolution, such as inheritance, mutation, selection, and crossover. More precisely, the algorithm first generate a random population space for selection. Next, crossover process select multiple couples of individuals (parents) from the population to create two new individuals (offsprings) by swapping a part of their genes. Finally,

mutation process randomly changes some old genes for the introduction of new genes. These processes are repeated until a pre-defined number of iterations and finally, the best candidate is selected. This algorithm is also proved effective in the applications of bioinformatics, phylogenetics, computational science, engineering, economics, chemistry, manufacturing, mathematics, physics, pharmacometrics, etc. However, the evolution process of a pure natural-simulated genetic algorithm is time consuming, and can easily get trapped into a local optima. This is because GA always generates the new populations in a random way firstly, and then selects the best individuals according to the fitness function, which enormously reduces the chance to find a better individual in the next iteration which results in a very low convergence rate. Thus, within HCR-KMGA, we perform a PFOA optimization process, which can raise the convergence rate by predicting the possible evolution directions.

Fig. 1 illustrates the over-all architecture of PFOA. Like the conventional GA, the system first initializes randomly the population. However, in the evolution process, only best-individual selection process are kept, while crossover-mutation and random selection are replaced by predictive evolution and random evolution. After each iteration, the best individuals are directly copied from the last generation into the next one for the purpose of fast convergence. Meanwhile, some of the individuals evolve depending on a prediction strategy, which can greatly further raise the convergence rate of population evaluation. Finally, the rest individuals are re-performed randomly in order to reduce the possibility of falling down to the local optima.

Depending on different fitness functions, designers can customize different prediction strategies. In our case, an individual has five skin tissue parameters. Their value ranges are displayed in Table 2. In KMGA, the size of population consist of a few hundred individuals, and in each iteration, several new genes are generated via crossover-mutation process. However, with the floating-point number, which is one of the most frequently used data format in computer science, KMGA has more than $6E27$ candidates,¹ which may result in a long running time and a very low convergence rate.

For the purpose of accelerating the convergence speed of algorithm, a prediction strategy that can reduce each iteration's search space by predicting the evaluation direction is performed as shown in the right of Fig. 1. Firstly, the best individuals of the last two generations are compared, and depending on the comparison result, the algorithm takes different steps to adjust the search space. We base the prediction strategy on the assumption that higher parameter values had better fitness while $x_{n-1} > x_{n-2}$, and lower parameter values had better fitness while $x_{n-1} < x_{n-2}$, where x_n refers to the parameter value of the best individual of the n th generation. As shown in Fig. 2(a) and (b), PFOA locks the search space U onto the scope of $x > x_{n-1}$ with $x_{n-1} > x_{n-2}$, while the scope of $x < x_{n-1}$ with $x_{n-1} < x_{n-2}$. Meanwhile, we note that this method is effective only for the situations that the present best individual is enough far away from the optima. Once it has been very closed to the optima, a much smaller search space may be required to enable the algorithm to find a better individual with as few iterations as possible. This situation is abstracted as $x_{n-1} = x_{n-2}$ and $f(x_{n-1}) = f(x_{n-2})$ in PFOA. It means that no better individuals are found in the last two iterations. Therefore, the search space of the n th iteration will be locked within the scope around x_{n-1} in order to enhance the chance of evolution (see Fig. 2(c)).

Since the optimization function is unknown, it is impossible to correctly predict the position of the global optima always. But this

¹ This value is estimated according to the range of the five skin parameters shown in Table 5.1 of [9] and the precision of the floating-point numbers.

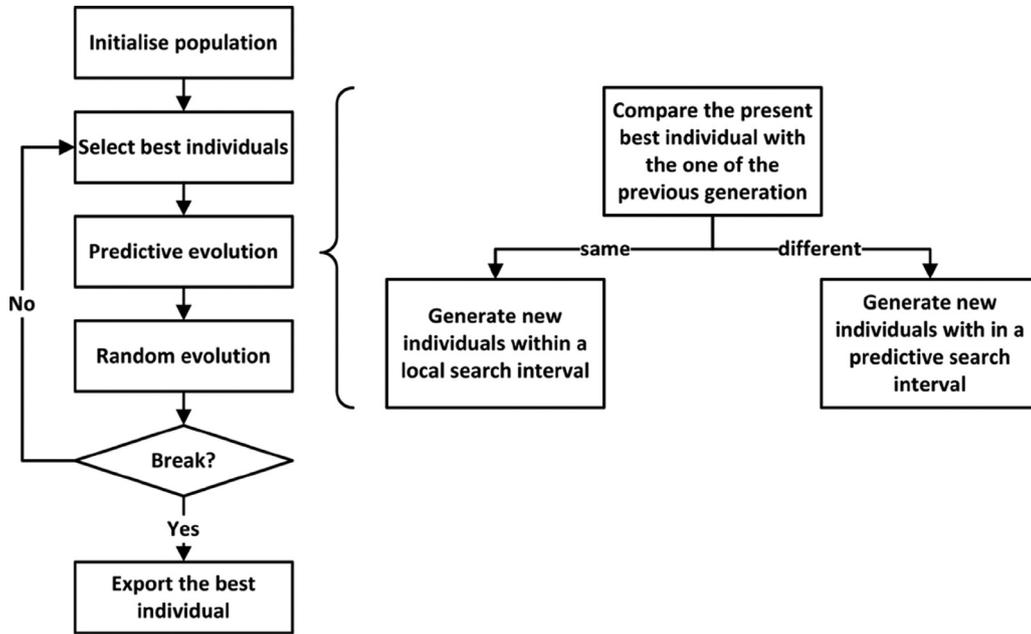


Fig. 1. Over-all architecture of PFOA.

Table 2
Size of search spaces for skin parameters.

Skin parameter	Symbol	Range
Melanin concentration	f_{met}	1.3–43%
Epidermis thickness	D_{epi}	0.01–0.15 mm
Volume blood fraction	f_{blood}	0.2–7%
Oxygen saturation	C_{oxy}	25–90%
Dermis thickness	D_{derm}	0.6–3 mm

mistake can be quickly corrected in the next iterations. For example, the predicting scope does not include the optima in Fig. 2(b), and within this scope no better individual can be found. However, this makes the algorithm restricts its search space around x_{n-1} in the following iterations, within which a new best individual can be easily found in the right of x_{n-1} .

In our case, KM model has five parameters to be figured out (see Table 1). Considering that these parameters have different effects to the final fitness, their analysis must be independent with each other. Thus, HCR-KMGA applies PFOA to all of them respectively. That is, after the fitness comparison, the search space of

each parameter is defined independently via the proposed prediction strategy.

It should also note that sometimes this method may also lead the evolution down to a local optima. Thus, after prediction evolution, some random individuals are performed in order to avoid it. Unlike GA, PFOA completely regenerates all the individuals in a random way instead of crossovers or mutations. This method greatly enriches the sample types of genes, so the risks of missing the optima is reduced.

2.3. Individual information storage optimization

In KMGA, the information of an individual consists of fitness value, chemical properties (the five skin parameters mentioned in Table 1) and optical properties (simulated spectrum). These data need to be saved in the memory of the processing device permanently through the whole processing, and it results that conventional KMGA prototype has to consume a lot of hardware resources to store all the population information, especially for embedded devices. Thus, an approach that can reduce memory consumptions is required.

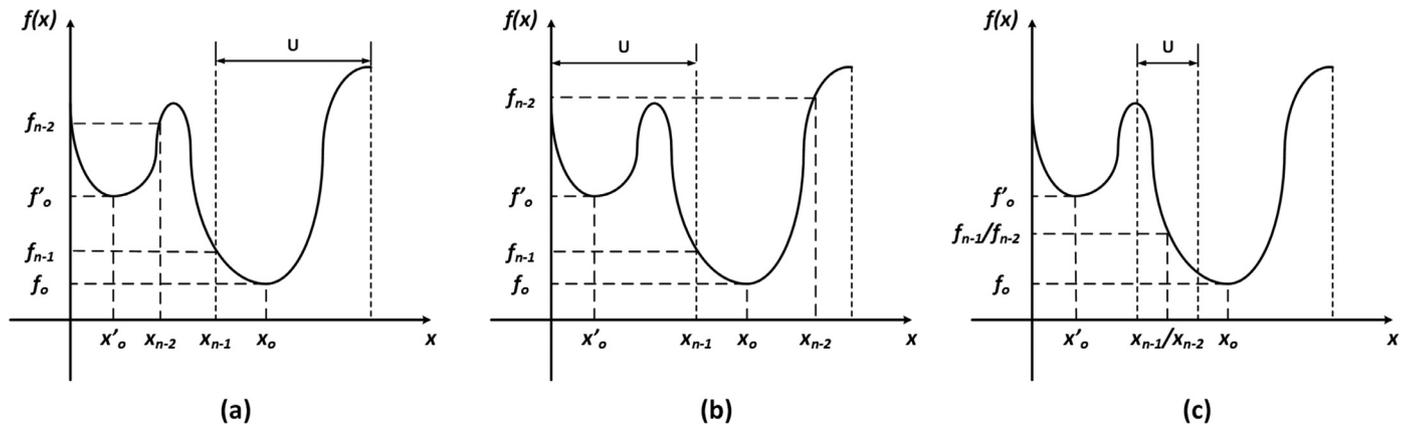


Fig. 2. Search space prediction of PFOA: x_n and f_n are the parameter and fitness value of the n th iterations best individual, (x'_o, f'_o) is the local optima and (x_o, f_o) is the global optima of the optimization function.

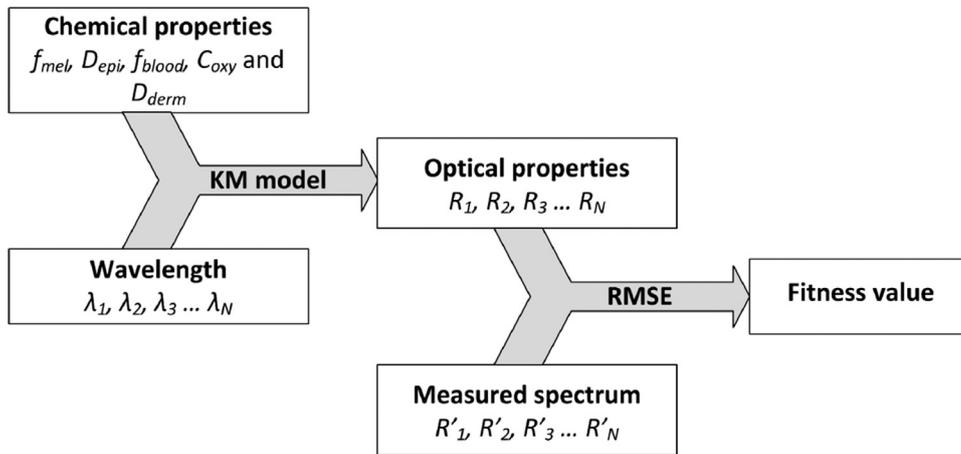


Fig. 3. Relationships of individual data: RMSE refers to Root Mean Squared Error.

According to the algorithm analysis, it is found that each piece of individual information is not isolated, but rather has internal relations with the others. Fig. 3 displays the relationships among them, and it demonstrates that both the simulated optical properties and the fitness value are calculated from the chemical properties. That provides a nice opportunity to compress the data size by removing out the data that could be computed immediately and keeping only the necessities.

The individuals of HCR-KMGA are performed by chemical properties and fitness values, while the information of its simulated optical properties is removed. In the over-all algorithm, the simulated spectrum of the skin lesion zone is used only once in order to compute the fitness values for the best individual selection at the beginning of each iteration (see Fig. 1), it is therefore unnecessary to allocate a certain memory to store them. Meanwhile, a single fitness value takes up only several octets, while its computing has to call the KM models, which has a very long running time. Nevertheless, this data is required not only in the best individual selection process, but also in the prediction evolution process. Thus, this number is stored as part of the individual information in HCR-KMGA in order to accelerate the design by reducing the operations number. The data size comparison between KMGA and HCR-KMGA's individual could be expressed as follow:

$$\frac{L_{KMGA}}{L_{HCR-KMGA}} = \frac{6 + N_{spectrum}}{6} \quad (11)$$

where L_{KMGA} and $L_{HCR-KMGA}$ are the total data length of the two algorithms' individuals and $N_{spectrum}$ is the bands number of the spectral image. Obviously, HCR-KMGA consumes much less memory space than KMGA, and its total length is permanently 6 times of the defined number length. That is, the algorithm will not take any more hardware resources for the population information storage even for the spectral images with high band numbers.

2.4. Termination conditions

The evolution process of Genetic Algorithm is terminated after a number of iterations according to the termination conditions customized by designers. A main issue that always affects the selection of termination conditions is that: defining a condition easy to reach consumes fewer hardware resources but may reduce the accuracy performances of designs, while a hard condition may lead to extensive computational time, the algorithm can even be trapped into an infinite loop.

KMGA terminates the evolution process by defining an iteration number which corresponds to the convergence of the population. That is, the evolution stops when the iterations number reaches

the default value. This approach could provide an acceptable average fitness value for the processing of a standard skin lesion multi-spectral image. However, the computations of GA is full of all kinds of possibilities, so such a simple termination condition may lead to redundant iterations or unpredictable results. For example, the evolution process will not be terminated until it reaches the default iterations number even if the global optima has been found, while the evolution process may have been terminated before the fitness reaches the required level. Thus, instead of forcing the algorithm to end by setting a default iterations limitation, HCR-KMGA combines multiple termination conditions together, including max continuous invalid iteration level $\theta_{invalid.iter}$, fitness level $\theta_{fitness}$ and total iteration level $\theta_{total.iter}$.

In our works, an iteration in which a new best individual is found is called a valid iteration, otherwise an invalid one. Once a large number of continuous invalid iterations appear, it means that the present best individual has been very closed to the optima, and it is difficult or time-consuming to find another one for the algorithm. So in order to save the hardware resources, a max continuous invalid iteration level is defined to break the evolution loop while no new best individual is found during $\theta_{invalid.iter}$ iterations.

Normally, the goal of evolution is not to exactly figure out the optima. That is, a fitness error could be accepted in each processing. In HCR-KMGA, a fitness level refers to the acceptable fitness error. Once the fitness value of the present best individual is lower than $\theta_{fitness}$, the evolution loop could be broken as well.

Finally, in order to avoid that the algorithm trap into an infinite loop, a total iteration level is defined. It should note that these three termination conditions effects simultaneously on the algorithm, so no matter which one is reached, the iterations will be ended. This method can save as many hardware resources as possible, and meet the accuracy requirements of the design as well.

3. Implementation and optimizations

As one of the most popular computation platforms, FPGAs have been used in a wide variety of real-time image processing applications. For example, Yonghong [22] discusses an efficient FPGA-based intellectual property (IP) core designing methodology to implement real-time image processing application such as normalized product correlation (NProd) image matching algorithm, Bodereau et al. [23] proposes a new non-conventional technique based on fuzzy deconvolution for scattering center detection (F-SCD) and its FPGA implementation for real-time deployment in automotive collision avoidance application and Komuro et al.

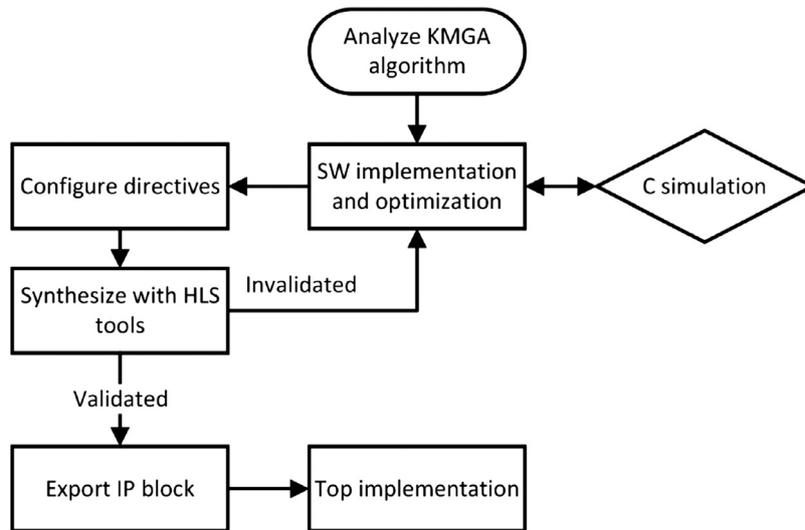


Fig. 4. Development process for the KMGA's FPGA implementation.

[24] develops an architecture of embedded systems for high-frame-rate real-time vision on the order of 1000 f/s, which achieved both hardware reconfigurability and easy algorithm implementation while fulfilling performance demands. In additional, many researches point out that FPGAs have demonstrable superiorities in terms of power-efficient compared with the other parallel and distribute platforms, such as multi-core CPUs or GPUs [15,25–27]. Depending on the proposed HCR-KMGA, a FPGA-based embedded skin lesion assessment system is implemented in Section 3.

3.1. Development framework

It is well known that FPGA is a reconfigurable device which has a quite different hardware architecture than the others. It needs to be configured in RTL before each use rather than run programs stored in the memories. This results that the achievements related to the other parallel and distribute platforms cannot be directly transplanted to it or even be referenced. For each design, software engineers have to transplant the desired algorithms from the original software environment (i.e. Matlab, C/C++ or OpenCV) into RTL or directly prototype the target implementations within RTL. Since RTL is a low-abstract development environment, both of these methods are very effort-consuming.

Table 3 Population parameters configurations for KMGA and HCR-KMGA.

Parameters	KMGA	HCR-KMGA
Population size	100	100
Best individuals	10	1
Random selection individuals	30	–
Crossing individuals	30 pairs	–
Mutation individuals	3	–
Prediction individuals	–	49
Random individuals	–	50
Maximum fitness plateau	–	6
Minimum satisfies fitness level	–	2E–4
Max generation number	25	80
Spectrum size	34	34

After a series of efforts related to the SW/HW Co-design framework for FPGAs, a novel HLS procedure, which is also known as C Synthesis, is used in our case. HLS is a source-to-source compilation method that can synthesize the C-like languages into the textual description of a circuit diagram or schematic for FPGA devices [16]. It provides a software-friendly hardware development environment, which keeps the coding restriction due to the hardware constraints down to minimum. Moreover, experienced C language

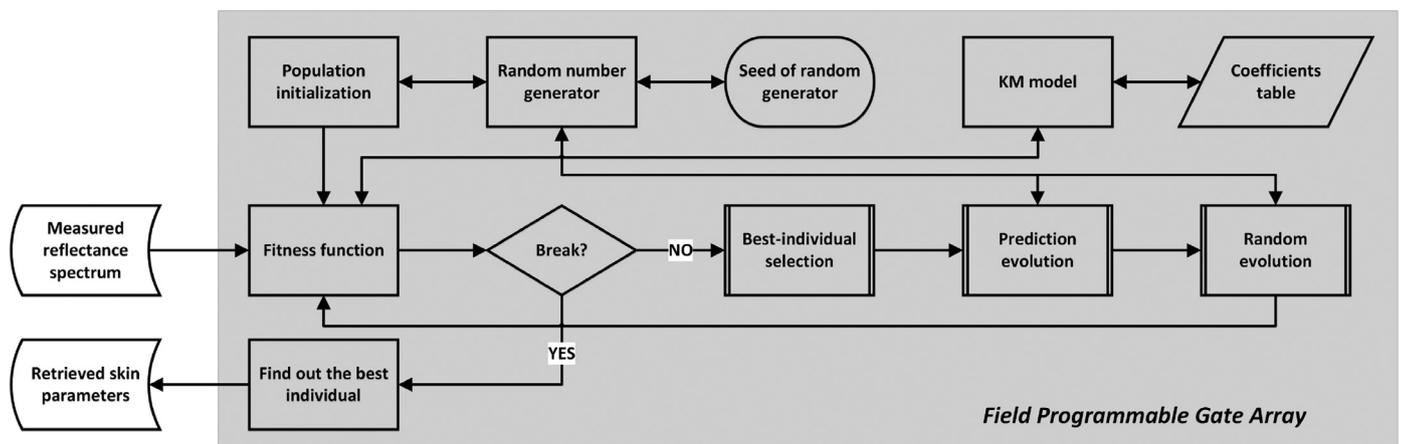
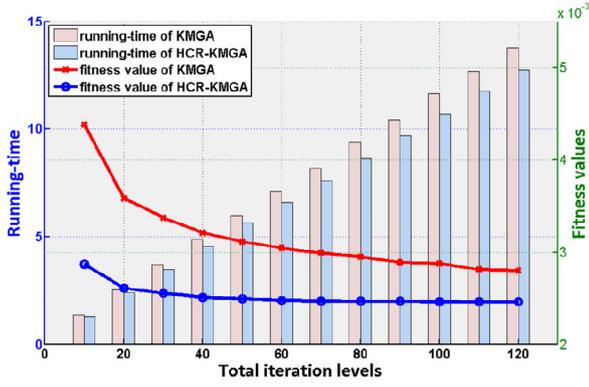
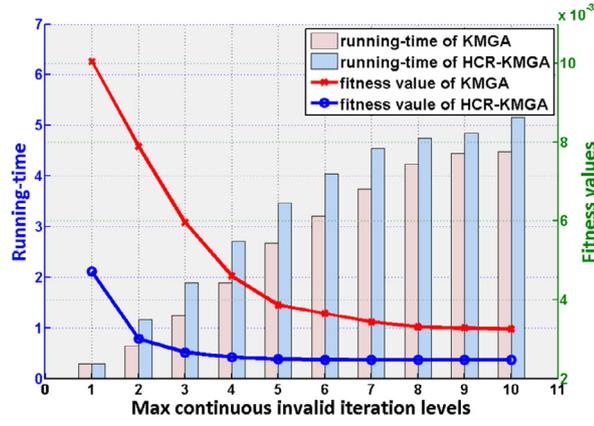


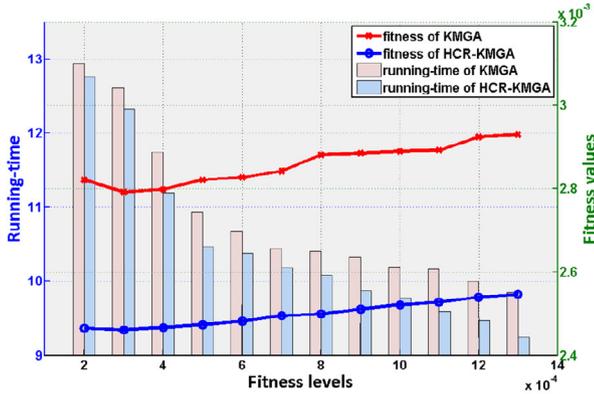
Fig. 5. Software implementation of HCR-KMGA algorithm for the FPGA device.



(a) Comparison between KMGA and HCR-KMGA with different total iteration levels: $\theta_{invalid.iter} = \infty$ and $\theta_{fitness} = 0$.



(b) Comparison between KMGA and HCR-KMGA with different max continuous invalid iteration levels: $\theta_{total.iter} = \infty$ and $\theta_{fitness} = 0$.



(c) Comparison between KMGA and HCR-KMGA with different fitness levels: $\theta_{invalid.iter} = \infty$ and $\theta_{total.iter} = 200$.

Fig. 6. Experimental results of KMGA vs. HCR-KMGA with different termination conditions: the definition of multi-spectral image is 36×30 , and the fitness values mentioned here are the average value of the pixels in it.

users only need to avoid to use the expressions unsupported by HLS tools in the source codes rather than learn a new tool.

Fig. 4 illustrates the HLS based development process used for our implementation. After a careful algorithm analysis, we first realize a synthesizable KMGA prototype in C language. Next, the prototype is optimized by using the approaches discussed in Section 2 and simulated through a common C compiler, i.e. Intel C++

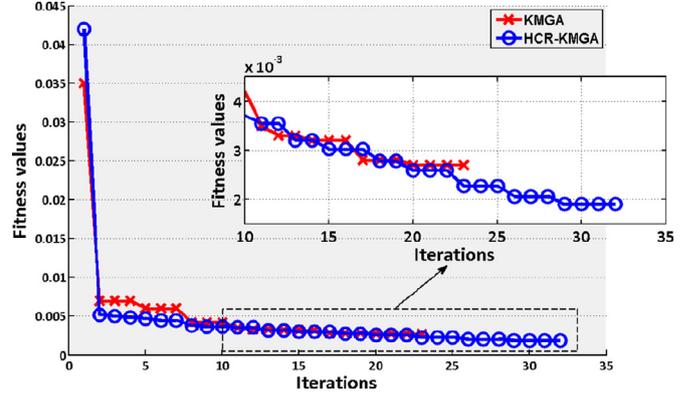


Fig. 7. Convergence rate comparison between KMGA and HCR-KMGA with a max continuous invalid iteration level of 4.

Compiler (ICC). Thirdly, the functionally verified source code is imported into the HLS tool for C-to-RTL synthesis. During this process, the synthesis directives are carefully configured in order to effectively use the hardware resources of the target device. Finally, the IP block of HCR-KMGA is automatically performed for top behavior.

3.2. Implementation

The over-all architecture of HCR-KMGA is shown in Fig. 5. Its initial population is generated according to the skin parameters' value range presented in Table 2. Giving that the experimental data acquisition devices is ASCLEPIOS (Analysis of Skin Characteristics by Light Emission and Processing of Image of Spectrum), whose waveband varies from 450 to 780 nm with the step of 10 nm, we replace the calculation of $\mu_{a,mel}$, $\mu_{a,baseline}$, $\mu_{a,oxy}$, $\mu_{a,deoxy}$ and $\mu_{s,epidermis}$ with different wavelengths by a coefficient table precalculated using the Takatani–Graham table [28]. As there is no golden rule to define the values of evolution process parameters, we base HCR-KMGA's parameters configuration on the results of intensive tests. Table 3 illustrates the evolution parameters of KMGA implemented by Jolivot et al. [13] and the proposed implementation. In the fitness computation, Root Mean Squared Error (RMSE) is applied as the metric scale to compare the candidates' simulated optical properties with the reference reflectance spectrum and express the fitness values. PFOA's evolution process consist of best-individual selection, prediction evolution and random evolution process. Moreover, in order to make the routine synthesizable, the standard C function *rand()* which is not supported by our HLS tool due to the use of static variable is replaced by a linear congruential random number generator coded manual.

Within the process of directive configuration, variant operation scheduling strategies could be effected in different hierarchies [17], such as function-level, loop-level and instruction-level parallelism strategies that the most frequently used for high performance computing. Nevertheless, this process can be kept repeating until the most optimal solution is found. In our case, we completely pipeline the fitness computation and the random number generator, while the function optimizer is partly pipelined due to the hardware constrains.

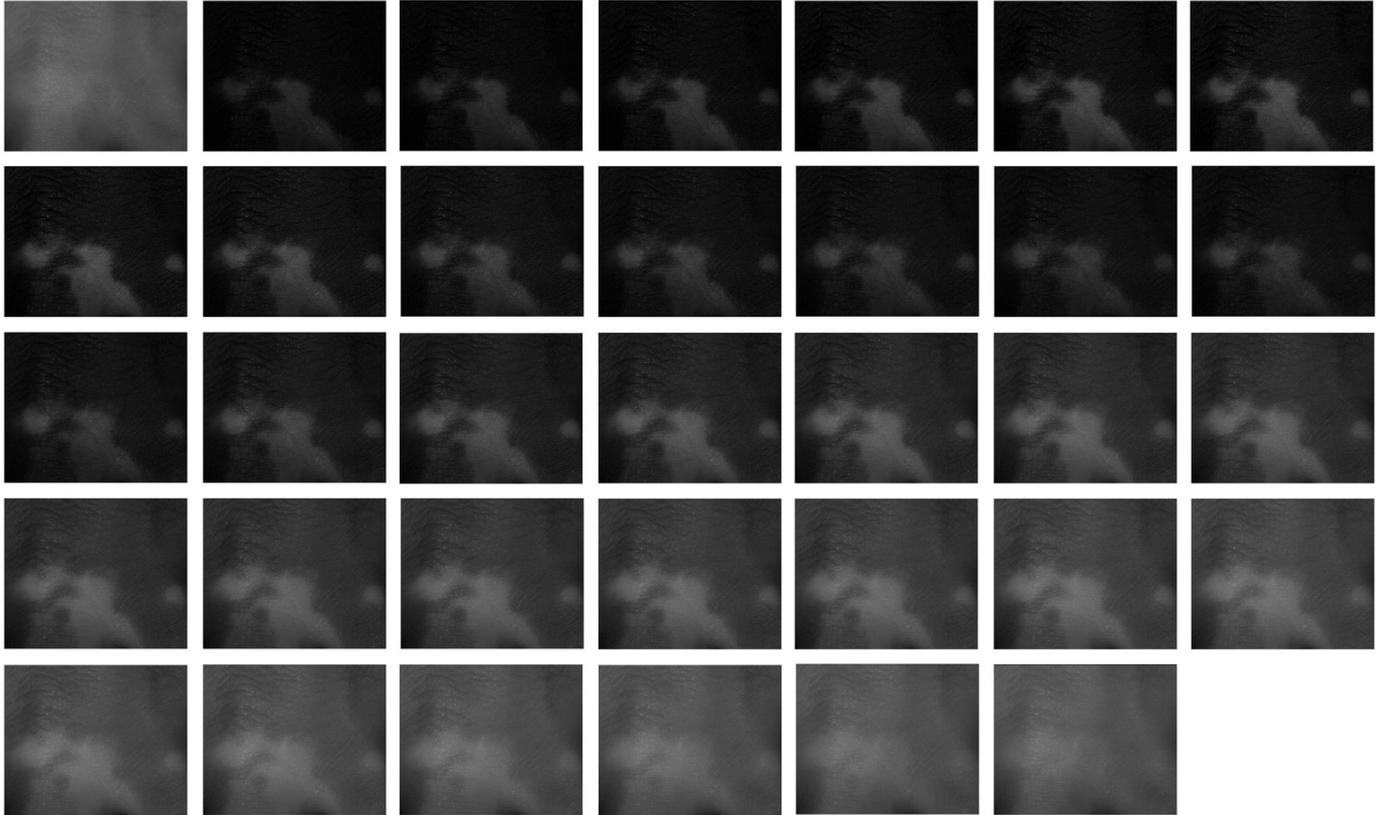
4. Experiments and analysis

4.1. Algorithm evaluation

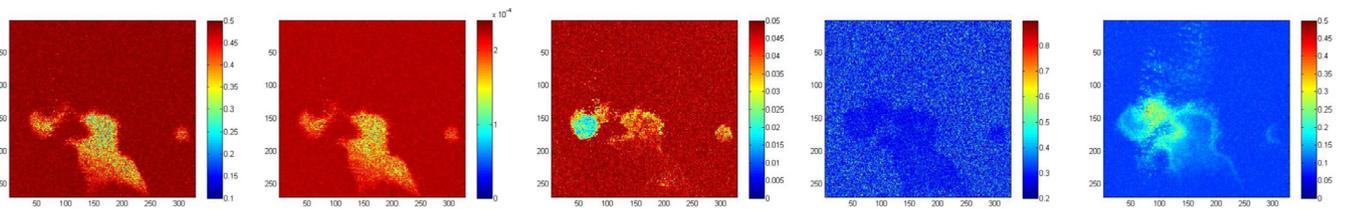
In this experiment, we evaluate the convergence rate of HCR-KMGA by comparing it with KMGA. Fig. 6 displays the experimental results of KMGA vs. HCR-KMGA with different termination

Table 4
Specifications of different KMGAs implementations: '*' refers to the implementation proposed by this paper.

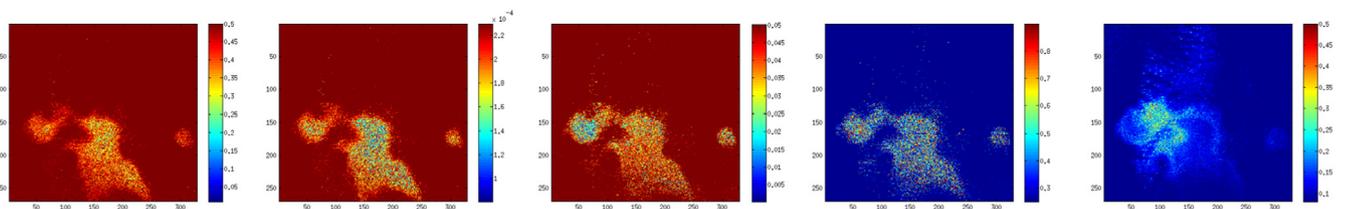
Implementations	Algorithms	Environments	Tools	Devices	Optimizations
KMGA	KMGA	C/C++	ICC	CPU Q6600	POSIX Threads
HCR-KMGA	HCR-KMGA	C/C++	ICC	CPU Q6600	POSIX Threads
FPGA-KMGA	KMGA	C/C++	Vivado_HLS	Virtex7	HLS directives
FPGA-HCR-KMGA*	HCR-KMGA	C/C++	Vivado_HLS	Virtex7	HLS directives



(a) Multispectral image measured by ASCLEPIOS from 450 to 780 nm with the step of 10 nm.



(b) Skin parameter maps retrieved by KMGA : melanin concentration map, epidermis thickness map, volume blood fraction map, oxygen saturation map and dermis thickness map ($\theta_{invalid.iter} = 25$).



(c) Skin parameter maps retrieved by HCR-KMGA : melanin concentration map, epidermis thickness map, volume blood fraction map, oxygen saturation map and dermis thickness map ($\theta_{invalid.iter} = 6$, $\theta_{fitness} = 2 \times 10^{-4}$ and $\theta_{total.iter} = 80$).

Fig. 8. Multi-spectral image measured by ASCLEPIOS and simulation results of KMGA and HCR-KMGA.

conditions. First of all, this result demonstrates that HCR-KMGA has a higher convergence rate than KMGA no matter with which termination condition (lower fitness values are better). This is because PFOA could effectively increase the chance to find a better individual and avoid to trap into a local optimal by reducing the search space according to the prediction strategies proposed in Section 2.2, while KMGA requires more iterations to reach the same fitness level.

On the other hand, what is interesting is that HCR-KMGA consumes more times than KMGA with the max continuous invalid iterations, while the other two sub figure indicate that the former should have a better efficiency performance. Indeed, if we assume that the two algorithms consume the same time for a single iteration, then HCR-KMGA should have been more efficiency in theory because it converges faster. However, actually higher convergence rate does not necessarily lead to lower running-time for all the three termination conditions. Fig. 7 displays a convergence rate comparison between KMGA and HCR-KMGA with a max continuous invalid iteration level of 4. It indicates that KMGA reaches the termination condition in the 23th iteration, while HCR-KMGA terminates in the 32th. It is known that once a new valid iteration appears, the algorithm has to reset its continuous invalid iterations count. Because of high convergence rate, HCR-KMGA could create a valid iteration more easily than KMGA before the algorithm reach the termination condition. That is, the former is able to play a longer evolution than the latter with the same $\theta_{invalid.iter}$ and retrieve a set of skin parameters more exact.

It should note that in the test of Fig. 6(c), in order to prevent the evolution from trapping into an infinite loop, a fixed total iteration level of 200 is defined as well as the fitness level according to Fig. 6(a), in which the fitness values of the both algorithms don't vary a lot after one hundred and twenty iterations.

Unlike the function reducing method used in our previous work, PFOA is more universal, because it optimizes the design by its nature instead of merely reducing the operation numbers without algorithm optimizing. It should be noted as well that PFOA is an artificial variety of evolutionary algorithm, and the potential pitfalls of human intervention to such applications are still open issues in the circle of science. Up to our knowledge, it does not exist yet a theoretically effective solution for them. In our case, half of the individuals are set as random individuals to prevent the evolution from falling into local optimal, while the medically-measured skin parameter ranges are applied to ensure the rationality of assessment results. Intensive experiments (328×270 multi-spectral image pixel samples) demonstrate that the simulated skin parameters retrieved through PFOA have lower fitness values from whole, and the maps generated from them have a better visual effect than the ones evolved according to the pure GA. Therefore, it is reasonable to consider this new proposed method feasible to the target issue by far.

4.2. Comparison experiment

This subsection evaluates proposed KMGA skin lesion assessment by comparing four different KMGA based skin lesion assessment implementations (specified in Table 4). In order to obtain an unbiased conclusion, the optical models of all the implementations perform the rewritten KM model introduced in Section 2.1. Furthermore, the algorithms are specified all in C/C++ language. That is, these implementations are developed from the same prototype. Since the target CPU device is a quad-core processor, we optimize the CPU implementations within a POSIX Thread framework. This method allows to multiply designs' efficiency by simultaneously running multiple threads depending on the core number of the target device [29].

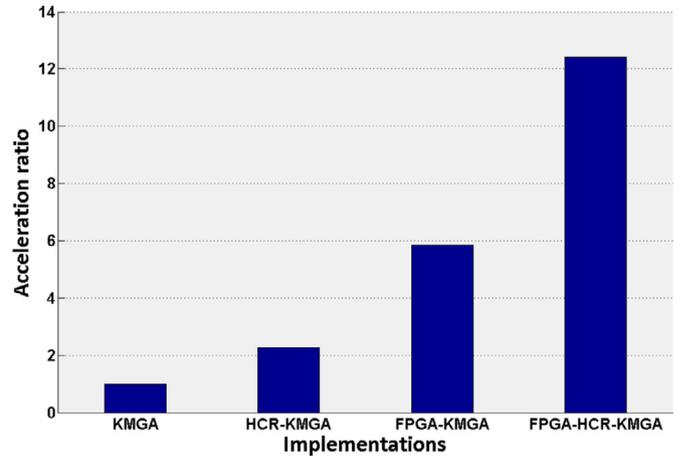


Fig. 9. Efficiency comparison: the clock frequencies of CPU and FPGA are respectively 2.4 GHz and 50 MHz ($\theta_{invalid.iter} = 6$, $\theta_{fitness} = 2 \times 10^{-4}$ and $\theta_{total.iter} = 80$).

Table 5

Hardware estimation of FPGA-KMGA and FPGA-HCR-KMGA on Virtex7-XC7VX1140T of Xilinx.

Components	FPGA-KMGA	FPGA-HCR-KMGA
BRAM_18K	192	32
DSP48E	2352	2431
FF	467264	493177
LUT	668784	712894

Fig. 8(b) and (c) display the skin parameter maps retrieved by KMGA and HCR-KMGA from a standard multi-spectral image shown in Fig. 8(a). Obviously, the latter makes less noise from the visual effects and appears a clearer skin lesion zone for diagnostics. The accuracy performances of the designs is quantitatively evaluated using the fitness value. We calculate the accuracy performance data of the CPU implementations from the matrixes of the retrieved skin parameter maps, and the FPGAs' are obtained through a test bench implemented using Simulink/SysGen. According to the test results, we can find that both the HCR-KMGA implementations has lower fitness values than the KMGA ones (2.6×10^{-4} vs. 3.4×10^{-4} and lower is better). Therefore, it could be concluded that the proposed algorithm performs better in term of accuracy no matter what hardware platform is used. Meanwhile, it is also found that the fitness values are almost identical to the FPGA and CPU implementations. This is because the HLS based SW/HW Co-design framework that we followed can well transplant an algorithm specified in C-like languages onto FPGAs almost without any omissions of functions.

The efficiency performances of our implementations are as well compared in Fig. 9. Thanks to the prediction evolution strategy, HCR-KMGA offers an acceleration gain of $2.28 \times$ relative to KMGA, while a gain of $2.13 \times$ for FPGA-HCR-KMGA vs. FPGA-KMGA. Nevertheless, the loop-level and instruction-level parallelism enable FPGAs to appear a much better hardware performances than CPUs, although it has a lower clock frequency. The speed gains due to the platform are $5.84 \times$ and $5.45 \times$ for FPGA-KMGA vs. KMGA and FPGA-HCR-KMGA vs. HCR-KMGA respectively.

Finally, we compare the hardware resources consumption of the two FPGA implementations in Table 5. This comparison indicates that HCR-KMGA consumes much less RAM than KMGA. This is because the data size of population are well reduced according to the approach presented in Section 2.3, it need not therefore to allocate as much storage space as before. In contrast, HCR-KMGA consumes more other components relative to RAM. This is because POFA has a more complex architecture than GA, HLS has

to spend more resources for the operation control flow. However, this difference is very tiny, it can almost be ignored in practical applications.

5. Conclusion

This paper presents a real-time embedded application of multi-spectral medical image processing technique. The conventional implementation is improved both in terms of software and hardware using multiple approaches.

Firstly, the new-proposed implementation is developed basing on a reduced optical model, which offers a high speed gain by reducing the necessary operations numbers during the computations. Then, the evolution process for this model is redesigned according to a prediction evolution strategy. This method greatly accelerates the algorithm by raising its convergence rate. Finally, the data relationships of individuals are re-analyzed in order to compress the individual information storage.

In additional, our design is implemented and evaluated within a High-Level Synthesis SW/HW Co-design framework. This method enable us to synthesize the C-specification of the algorithms directly into RTL for FPGAs. Thus, all the experiment analysis of this paper effect in a software-friendly environment rather than a hardware-friendly one, which could free the software engineers from the boring and insignificant hardware specifications, and put more effort in the algorithm improvement. Our work demonstrates that this framework could greatly accelerate the period of SW/HW Co-designs.

The design improvement of this work is based on a reduced KM function previously developed in [21]. The new proposed implementation is further accelerated in both algorithm and hardware aspects. According to the experiments discussed in Section 4, the improved algorithm has a higher convergence rate than its conventional prototype and a better accuracy as well. This achieves a speedup of about $2 \times$ in algorithm aspect. Meanwhile, the comparison between CPU and FPGA implementations demonstrates that the latter can accelerate the design by about $6 \times$ in hardware aspect. Totally, the proposed embedded design provides a speed gain of around $12 \times$.

In the future works, we plan to further optimize the proposed design's performance from two aspects:

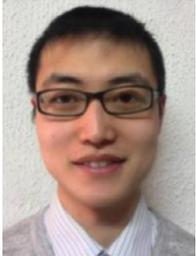
- (a) Attempt to transplant KMGA onto GPUs. Many comparative research point out GPUs may achieve more potential efficiency improvements. However, our pre-experiments demonstrate that KMGA is not amenable to the currently-available GPUs, i.e. TESLA C2050 of NVIDIA, due to the constraints of local memory capacity. Therefore, the over-all architecture of the algorithm may be re-designed probably for this purpose.
- (b) Finding a more efficient operations scheduling strategy within the C-to-RTL synthesis process. After the effort of this paper, its experiment results and the ones obtained during the other related work will be analyzed. The final goal is to find a general fast FPGA development flow for real-time image processing designs. We believe that it still exists many interesting challenges and opportunities that may offer more potential performance optimizations to the embedded real-time image processing designs,

Finally, we hope that the achievements of this work can bring some new enlightenments to the related researches.

References

- [1] R. Jolivot, Y. Benezeth, F. Marzani, Skin parameter map retrieval from a dedicated multispectral imaging system applied to dermatology/cosmetology, *Int. J. Biomed. Imaging* 2013 (2013) 15, doi:10.1155/2013/978289.
- [2] P. Kubelka, F. Munk, Ein Beitrag zur optik der farbanstriche, *Z. Tech. Phys.* 9 (1931) 593–601.
- [3] R. Jolivot, Development of an imaging system dedicated to the acquisition, analysis and multispectral characterisation of skin lesions, University of Burgundy, 2011 (Ph.D. thesis).
- [4] J. Jiang, C. Liu, S. Ling, An FPGA implementation for real-time edge detection, *J. Real-Time Image Process.* (2015) 1–11, doi:10.1007/s11554-015-0521-7.
- [5] A. Senturk, M. Gok, Sequential large multipliers on FPGAs, *J. Signal Process. Syst.* 81 (2) (2015) 137–152, doi:10.1007/s11265-014-0912-1.
- [6] S. Musavi, B. Chowdhry, T. Kumar, B. Pandey, W. Kumar, lots enable active contour modeling based energy efficient and thermal aware object tracking on fpga, *Wirel. Pers. Commun.* 85 (2) (2015) 529–543, doi:10.1007/s11277-015-2753-z.
- [7] B. Sukhwani, M. Thoennes, H. Min, P. Dube, B. Brezzo, S. Asaad, D. Dillenberger, A hardware/software approach for database query acceleration with fpgas, *Int. J. Parallel Progr.* 43 (6) (2015) 1129–1159, doi:10.1007/s10766-014-0327-4.
- [8] C. Colodro-Conde, F.J. Toledo-Moreo, R. Toledo-Moreo, J.J. Martinez-Ivarez, J.G. Guerrero, J.M. Ferrndez-Vicente, Evaluation of stereo correspondence algorithms and their implementation on fpga, *J. Syst. Archit.* 60 (1) (2014) 22–31. <http://dx.doi.org/10.1016/j.sysarc.2013.11.006>.
- [9] H. Sidiropoulos, K. Siozios, D. Soudris, A novel 3-d FPGA architecture targeting communication intensive applications, *J. Syst. Archit.* 60 (1) (2014) 32–39. <http://dx.doi.org/10.1016/j.sysarc.2013.09.012>.
- [10] F.J. Toledo-Moreo, J.J. Marnez-Alvarez, J. Garrigs-Guerrero, J.M. Ferrndez-Vicente, FPGA-based architecture for the real-time computation of 2-d convolution with large kernel size, *J. Syst. Archit.* 58 (8) (2012) 277–285. <http://dx.doi.org/10.1016/j.sysarc.2012.06.002>.
- [11] S. Lyberis, G. Kalokerinos, M. Lygerakis, V. Papaefstathiou, I. Mavroidis, M. Kat-evenis, D. Pneumatikatos, D.S. Nikolopoulos, FPGA prototyping of emerging manycore architectures for parallel programming research using formic boards, *J. Syst. Archit.* 60 (6) (2014) 481–493. <http://dx.doi.org/10.1016/j.sysarc.2014.03.002>.
- [12] M. Baklouti, Y. Aydi, P. Marquet, J. Dekeyser, M. Abid, Scalable mpnoc for massively parallel systems – design and implementation on FPGA, *J. Syst. Archit.* 56 (7) (2010) 278–292. <http://dx.doi.org/10.1016/j.sysarc.2010.04.001>. Special Issue on HW/SW Co-Design: Systems and Networks on Chip.
- [13] D. Zou, Y. Dou, F. Xia, Optimization schemes and performance evaluation of smith-waterman algorithm on CPU, GPU and FPGA, *Concurr. Comput. : Pract. Exper.* 24 (14) (2012) 1625–1644, doi:10.1002/cpe.1913.
- [14] S. Kestur, J. Davis, O. Williams, BLAS comparison on FPGA, CPU and GPU, in: Proceedings of 2010 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), 2010, pp. 288–293, doi:10.1109/ISVLSI.2010.84.
- [15] C. González, S. Sánchez, A. Paz, J. Resano, D. Mozos, A. Plaza, Use of FPGA or GPU-based architectures for remotely sensed hyperspectral image processing, *Integr. VLSI J.* 46 (2) (2013) 89–103. <http://dx.doi.org/10.1016/j.vlsi.2012.04.002>.
- [16] J. Cong, B. Liu, S. Neuendorffer, J. Noguera, K. Vissers, Z. Zhang, High-level synthesis for FPGAs: From prototyping to deployment, *IEEE Trans. Computer-Aided Des. Integr. Circuits Syst.* 30 (4) (2011) 473–491, doi:10.1109/TCAD.2011.2110592.
- [17] Y. Liang, K. Rupnow, Y. Li, D. Min, M.N. Do, D. Chen, High-level synthesis: productivity, performance, and software constraints, *J. Electr. Comput. Eng.* 2012 (2012) 14, doi:10.1155/2012/649057.(Article ID 649057)
- [18] L. Svaasand, L. Norvang, E. Fiskerstrand, E. Stopps, M. Berns, J. Nelson, Tissue parameters determining the visual appearance of normal skin and port-wine stains, *Lasers Med. Sci.* 10 (1) (1995) 55–65, doi:10.1007/BF02133165.
- [19] J.M. Schmitt, G.X. Zhou, E.C. Walker, R.T. Wall, Multilayer model of photon diffusion in skin, *J. Opt. Soc. Am. A* 7 (11) (1990) 2141–2153.
- [20] A.D. Kim, M. Moscoso, Light transport in two-layer tissues, *J. Biomed. Opt.* 10 (3) (2005) 034015, doi:10.1117/1.1925227.(Jun. 1 2015)
- [21] C. Li, V. Brost, Y. Benezeth, F. Marzani, F. Yang, Design and evaluation of a parallel and optimized light-tissue interaction based method for fast skin lesion assessment, *Journal of Real-Time Image Processing*, Springer Berlin Heidelberg, 2015, pp. 1–14, doi:10.1007/s11554-015-0494-6.
- [22] Z. Yonghong, An NProd algorithm IP design for real-time image matching application onto FPGA, in: 2010 International Conference on Electrical and Control Engineering (ICECE), 2010, pp. 404–409, doi:10.1109/ICECE.2010.105.
- [23] F. Bodereau, L. Giubolini, P. Mallejac, A. Pizzardi, 2D image fuzzy deconvolution and scattering centre detection: model and real-time FPGA implementation for automotive application, in: 2010 European Radar Conference (EuRAD), 2010, pp. 427–430.
- [24] T. Komuro, T. Tabata, M. Ishikawa, A reconfigurable embedded system for 1000 f/s real-time vision, *IEEE Trans. Circ. Syst. Vid. Technol.* 20 (4) (2010) 496–504, doi:10.1109/TCSVT.2009.2035832.
- [25] D. Zou, Y. Dou, F. Xia, Optimization schemes and performance evaluation of Smith-Waterman algorithm on CPU, GPU and FPGA, *Concurr. Comput. : Pract. Exper.* 24 (14) (2012) 1625–1644, doi:10.1002/cpe.1913.
- [26] S. Kestur, J. Davis, O. Williams, Blas comparison on FPGA, CPU and GPU, in: 2010 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), 2010, pp. 288–293, doi:10.1109/ISVLSI.2010.84.

- [27] J. Fowers, G. Brown, P. Cooke, G. Stitt, A performance and energy comparison of FPGAs, GPUs, and multicores for sliding-window applications, in: Proceedings of the ACM/SIGDA International Symposium on Field Programmable Gate Arrays, FPGA'12, ACM, New York, NY, USA, 2012, pp. 47–56, doi:[10.1145/2145694.2145704](https://doi.org/10.1145/2145694.2145704).
- [28] S.L. Jacques, [Spectroscopic determination of tissue optical properties using optical fiber spectrometer](#), Tech. Rep., Biomedical Engineering and Dermatology, Oregon Health and Science University (OHSU), Portland, Oregon, USA, 2008.
- [29] ISO/IEC, ISO/IEC 9945-1. ANSI/IEEE STD 1003.1. Information technology – portable operating system interface (POSIX). Part 1: system application interface (API) [C language], 1996.



Chao Li received the M.S. in Electronic Engineering from the University of Burgundy in 2012 and the M.S. in Computer Science from the University of Paris Sud in 2013, respectively. He is currently a Ph.D. candidate of LE2I CNRS-UMR, Laboratory of Electronic, Computing and Imaging Sciences at the University of Burgundy. His research interests are in the areas of SW/HW Co-design, High Performance Computer and Image Processing.



Souleymane Balla-Arabé received the M.Eng. degree in electronic from the Polytechnic Military School of Algiers in 2004, and the Ph.D. degree in information and communication engineering from Xidian University, Xi'an, China, in 2014 with distinction 'Excellent'. He is currently a postdoctoral research fellow at the LE2I CNRS-UMR, Laboratory of Electronic, Computing and Imaging Sciences, University of Burgundy. His research interests are architecture-aware optimization strategies for image segmentation, GPU-based computations, computational intelligence, machine learning, and computer vision. In these areas, he has published several technical articles in refereed journals and proceedings including IEEE Transactions on Geoscience and Remote Sensing, and the IEEE Transactions on Cybernetics.



Fan Yang received the B.S. degree in electrical engineering from the University of Lanzhou, China, in 1982 and the M.S. (D.E.A.) (computer science) and Ph.D. degrees (image processing) from the University of Burgundy, France, in 1994 and 1998, respectively. She is currently a full professor and member of LE2I CNRS-UMR, Laboratory of Electronic, Computing, and Imaging Sciences at the University of Burgundy, France. Her research interests are in the areas of patterns recognition, neural network, motion estimation based on spatio-temporal Gabor filters, parallelism and real-time implementation, and, more specifically, automatic face image processing algorithms and architectures. F. Yang is member of the French research group ISIS (Information, Signal, Images and Vision), she lives up the theme C: Algorithm Architecture Adequation.