SCALED FIXED-POINT FREQUENCY SELECTIVE EXTRAPOLATION FOR FAST IMAGE ERROR CONCEALMENT

Nils Genser, Jürgen Seiler, and André Kaup

Multimedia Communications and Signal Processing,
Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Cauerstr. 7, 91058 Erlangen, Germany
{nils.genser, juergen.seiler, andre.kaup}@fau.de

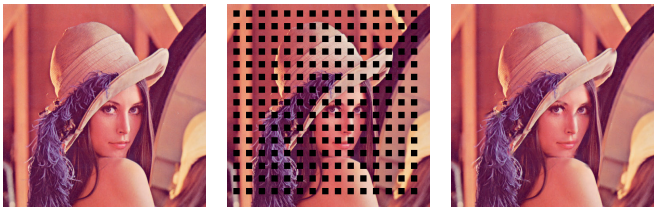
ABSTRACT

Image and video signal processing demands in many areas for error concealment algorithms, whereby the execution time plays an important role. Within this paper, we introduce the scaled fixed-point Frequency Selective Extrapolation for fast image error concealment. This algorithm is based on the complex-valued Frequency Selective Extrapolation, but reduces computational complexity. It is shown that a fixed-point realization of the state-of-the-art algorithm requires an impracticable word-length, which makes it necessary to design a novel, scaled Frequency Selective Extrapolation that reduces the required word-length. Regarding the new approach, execution time can be reduced on average by 22.84 % and at best by up to 40.45 % compared to the state-of-the-art floating point Frequency Selective Extrapolation at the same reconstruction quality. Moreover, platforms can be exploited, which support fixed-point arithmetic only.

Index Terms – Error Concealment, Inpainting, Signal Extrapolation

1. INTRODUCTION

An important task in image and video signal processing is the concealment of errors (see Fig. 1). There, typical examples are the concealment of transmission errors in wireless video communication [1] or the removal of objects from images, called inpainting [2]. Especially when time is a critical factor, greedy sparse algorithms are of particular interest [3]. They can produce a high quality, while saving computational resources using the fact that image signals are sparse in certain transform domains [4]. One of these effective algorithms is the Frequency Selective Extrapolation (FSE) working in the Fourier Domain [5]. In the past years, many improvements to the original FSE algorithm were introduced. While adaptions shown in [6] or [7] focuses on improving reconstruction quality, the approach presented in this paper accelerates algorithm execution. Moreover, alternative platforms, which do not support floating-point numbers, can be exploited. As a consequence, Field Programmable Gate Array (FPGAs) can be targeted now as they require fixed-point arithmetics for efficient implementations [8].



(a) (b) (c)

Fig. 1. Error concealment of block losses. (a) Original image. (b) Original image superimposed with error mask. (c) Concealed image using the proposed scaled fixed-point algorithm.

The complex-valued Frequency Selective Extrapolation, which was proposed in [9] serves as basis for the novel approach in this paper. While [9] focuses on reducing computational complexity by modifying the model generation, the novel approach decreases the computational effort even more by exploiting properties of fixed-point arithmetics. Performance gains inside the extrapolation loop sum up over large number of iterations, so that execution time can be reduced by up to 40.45 % as shown in Sec. 4.

2. COMPLEX-VALUED FREQUENCY SELECTIVE EXTRAPOLATION

The complex-valued Frequency Selective Extrapolation from [9] divides the image in areas of size $M \times N$ at first. Afterwards, signal $s[m,n]$ is recovered in block b using support area $\mathcal{A}$ and already reconstructed pixel $\mathcal{R}$. All together extrapolation area $\mathcal{L}$ is formed, which is of size $M \times N$ and depicted by the coordinates m and n. This classification is shown in Fig. 2 for an example of some reconstructed and lost pixels.

FSE generates the complex-valued model

$$g[m,n] = \sum_{k \in \mathcal{K}} \hat{c}_k \varphi_k[m,n] \tag{1}$$

by superimposing Fourier basis functions. Here, $\hat{c}_k$ names the expansion coefficient, $\varphi_k[m,n]$ the Fourier basis functions

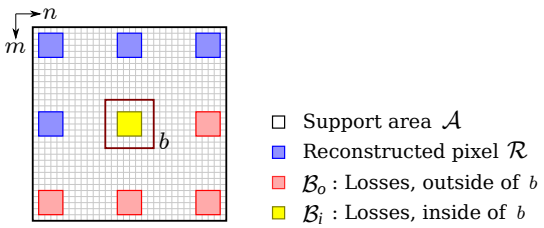


Fig. 2. Extrapolation area $\mathcal{L}$ containing support area $\mathcal{A}$, reconstructed pixel $\mathcal{R}$ and losses $\mathcal{B}_o$ and $\mathcal{B}_i$.

and $\mathcal{K}$ holds the indexes of all basis functions used for model generation. By adding a basis function in every new step, the model is generated iteratively. In addition, expansion coefficient $\hat{c}_k$ has to be estimated, which is described in [10].

The algorithm can be carried out in the frequency domain and only the transforms of the input signal, of the weighting function and another one back to spatial domain are required [9]. During modeling, the weighting function

$$w[m,n] = \begin{cases} \hat{\rho}^{\sqrt{\left(m-\frac{M-1}{2}\right)^2+\left(n-\frac{N-1}{2}\right)^2}} & , (m,n) \in \mathcal{A} \\ \delta\hat{\rho}^{\sqrt{\left(m-\frac{M-1}{2}\right)^2+\left(n-\frac{N-1}{2}\right)^2}} & , (m,n) \in \mathcal{R} \\ 0 & , (m,n) \in \mathcal{B}_i \cup \mathcal{B}_o \end{cases} \quad (2)$$

is used to control the influence of the pixels to the model generation. Here, the weighting function is used to model the influence of samples to the model generation depending on its distance to the distorted block b. Therefore, $\hat{\rho}$ controls the decay of the weighting function. Moreover, δ is used to weight already reconstructed pixels in $\mathcal{R}$ weaker than ones from support area $\mathcal{A}$. This is due to the fact that already reconstructed pixels are not as reliable as known ones. Also, inner and outer losses $\mathcal{B}_i$ and $\mathcal{B}_o$ get a zero weight to be masked from model generation process. After determining $w[m,n]$, it is transformed into the frequency domain to obtain $W[k,l]$ and the initial residual is calculated by performing

$$R_w[k,l] = FFT\{s[m,n] \cdot w[m,n]\} \; . \quad (3)$$

Here, k and l depict the indices in the frequency domain. Now, the FSE algorithm repeats for I times the process of basis function selection. Therefore, the maximum of the residual is determined by

$$(u^{(\nu)}, v^{(\nu)}) = \text{argmax} \, |R_w[k,l]|^2 \; , \quad (4)$$

while $(u^{(\nu)}, v^{(\nu)})$ names the coordinates of maximum at iteration ν. In the next step, the expansion coefficient

$$\hat{c}_{(u^{(\nu)},v^{(\nu)})} = \gamma \cdot R_w^{(\nu-1)}[u^{(\nu)}, v^{(\nu)}] \; / \; W[0,0] \quad (5)$$

is calculated using the orthogonality compensation factor γ [10]. In the following, the model

$$G^{(\nu)}[u^{(\nu)}, v^{(\nu)}] = G^{(\nu-1)}[u^{(\nu)}, v^{(\nu)}] + M \cdot N \cdot \hat{c}_{(u^{(\nu)},v^{(\nu)})} \quad (6)$$

is updated using coefficient from (5). Finally, the whole residual has to be brought up to date by performing

$$R_w^{(\nu)}[k,l] = R_w^{(\nu-1)}[k,l] - \hat{c}_{(u^{(\nu)},v^{(\nu)})} \cdot W[k-u^{(\nu)}, l-v^{(\nu)}], \forall (k,l) \; . \quad (7)$$

After performing all iterations, $\mathcal{B}_i$ is extracted from the generated model and used for extrapolation of the signal $s[m,n]$.

3. ALGORITHM ADAPTION AND OPTIMIZATION

The original algorithm from [9] is not suited to work with fixed-point arithmetic without using a high word-length for its calculations. Therefore, the required data-types and its disadvantages are discussed at first. Afterwards, the scaled FSE algorithm is proposed, which reduces the word-length, but leads to significant algorithm adaptions. Finally, the weighting function is investigated as it requires its own well-fitted data-type and functions like square-root and power approximation in order to be realized in fixed-point arithmetic.

3.1. Reconstruction

As already mentioned before, the proposed approach uses fixed-point arithmetic to obtain performance benefits. A word consisting of letters $c_i \in \{0,1\}$ with i being integers between $n-1$ to $-m$ leads to a binary representation

$$z_2 = c_{n-1} \; c_{n-2} \; ... \; c_0 \; ... \; c_{-m+1} \; c_{-m} \; . \quad (8)$$

For n pre-decimal and m decimal digits, one obtains the according number in the decimal system

$$z_{10} = c_{n-1} \cdot 2^{n-1} + ... + c_0 \cdot 2^0 + ... + c_{-m} \cdot 2^{-m} \quad (9)$$

by multiplying weights to letters respective position in its word. While addition and subtraction of fixed point numbers can be similarly conducted as with integers, one needs to pay attention performing multiplication and division [11]. Both lead to an increased word-length, due to the fact that either the number of digits for value range or accuracy grow. So, for both operations an intermediate word-length of the sum of both word-lengths has to be stored [11]. Afterwards, the original word-length is restored by shifting the intermediate result and dropping some accuracy.

Modern computers use words with length of power of two and support up to 64 bit [12]. Due to these circumstances, it is advisable to use 32 bit fixed-point variables, hence intermediate results do not exceed 64 bit. Calculations based on 64 bit word length would lead to 128 bit interim values, which should be avoided due to speed issues.

Since the state-of-the-art FSE requires a certain dynamic range for its calculations, at least a 64 bit fixed-point datatype is required. However, this would lead to 128 bit interim results and therefore to an increased execution time, which is

Alg. 1. The proposed scaled fixed-point FSE algorithm. The differences compared to the state-of-the-art (SoA) floating-point FSE are marked in green.

input: $s[m,n]$ and $w[m,n]$
/* *Scaling of the input signal* */
$s[m,n] = s[m,n] \,/\, (M \cdot N)$
/* *Transform signals into Fourier domain* */
$R_w[k,l] = \mathrm{FFT}\{s[m,n] \cdot w[m,n]\}$
$W[k,l] = \mathrm{FFT}\{w[m,n]\}$
$\overline{W}_{\gamma,o_{\hat{c}}} = \frac{(\gamma/o_{\hat{c}})}{W_{\mathrm{r}}[0,0]}$ $\left(\text{SoA: } \overline{W}_0 = \frac{1}{W[0,0]}\right)$
for all $\nu = 1, ..., I$ **do**
  /* *Basis function selection* */
  $(u,v) = \mathrm{argmax}_{(k,l)} |R_w[k,l]|^2$
  /* *Expansion coefficient estimation* */
  $\hat{c} = (R_w[u,v] \cdot \overline{W}_{\gamma,o_{\hat{c}}}) \cdot o_{\hat{c}}$ $\left(\text{SoA: } \hat{c} = \gamma \cdot R_w[u,v] \cdot \overline{W}_0\right)$
  /* *Model update* */
  $G[u,v] = G[u,v] + \hat{c}$ (SoA: $G[u,v] = G[u,v] + MN \cdot \hat{c}$)
  /* *Residual update* */
  for all $k = 0, ..., M-1 \wedge l = 0, ..., N-1$ **do**
    $R_w[k,l] = R_w[k,l] - \hat{c} \cdot W[k-u, l-v]$
  end for
end for
/* *Retransform model into spatial domain* */
$g[m,n] = \mathrm{IFFT}\{G[k,l]\} \cdot MN$ (SoA: $g[m,n] = \mathrm{IFFT}\{G[k,l]\}$)
/* *Replace distorted signal parts* */
for all $(m,n) \in \mathcal{B}$ **do**
  $s[m,n] = \mathrm{Re}\{g[m,n]\}$
end for
output: $s[m,n]$

unwanted. To overcome this problem, the scaled FSE algorithm is introduced in this paper, which uses a limited value range of $[-1;+1)$ for the input signal. Due to this, only a single pre-decimal digit has to be used, so that the needed word length can be reduced to 32 bit. As a consequence, the FSE algorithm must be adapted at several positions (see Alg. 1).

Starting from an image, where the pixels are stored as 8 bit words, a conversion to 32 bit fixed-point numbers is performed at first. By doing an adequate number of bit shifts the values are scaled, so that the input signal matches the range $[-1;+1)$. This representation offers the benefit that multiplications can not overflow. Moreover, additions and subtractions will not overflow, when kept in the range $[-1/2;+1/2)$.

Keeping these advantages in mind, the next step of algorithm adaption is investigated. Here, the weighting function and the residual have to be transformed to frequency domain. For reduced calculation complexity, an FFT is advisable. In the implementation shown afterwards, the KISS C library is used [13]. However, the transformation to frequency domain enlarges the input signal by a factor of the FFT window size $M \times N$. Due to this, the signal has to get downscaled by $M \times N$ to counteract.

The reduced dynamic of spectrum, makes algorithmic modifications essential compared to the original algorithm shown in [9]. Firstly, the calculation of the expansion coefficient from (5) requires a modification. There, the Fast Orthogonality Deficiency Compensation factor γ [10] is multiplied with the residual $R_w[u,v]$ and divided by the weighting function at position zero $W[0,0]$. As division can lead to an overflow, adjustments are required. Using the property, that the direct component part of a signal is real-valued only, for real and imaginary part

$$\hat{c}_{\mathrm{r}} = \gamma \cdot \frac{R_{w,\mathrm{r}}[u,v]}{W_{\mathrm{r}}[0,0]} \quad \text{and} \quad \hat{c}_{\mathrm{i}} = \gamma \cdot \frac{R_{w,\mathrm{i}}[u,v]}{W_{\mathrm{r}}[0,0]} \tag{10}$$

result. At this point, one has to pay attention that the denominator of (10) is bigger than the numerator. This is achieved by dividing γ by a weighting scaling constant $o_{\hat{c}}$ to obtain a smaller number than $W_{\mathrm{r}}[0,0]$. Now, the division is performed, which produces results in range $[-1;+1)$. In the next step, the multiplication with the residual is conducted and the division by $o_{\hat{c}}$ is reverted. To reduce computational complexity, the calculation can be performed in advance of the extrapolation loop since γ, $o_{\hat{c}}$ and $W_{\mathrm{real}}[0,0]$ do not change over iterations. As a result, the global weighting constant

$$\overline{W}_{\gamma,o_{\hat{c}}} = \frac{(\gamma/o_{\hat{c}})}{W_{\mathrm{r}}[0,0]} \tag{11}$$

is calculated before the loop. Afterwards, real and imaginary part of the expansion coefficient

$$\hat{c}_{\mathrm{r}} = R_{w,\mathrm{r}}[u,v] \cdot \overline{W}_{\gamma,o_{\hat{c}}} \cdot o_{\hat{c}} \quad \text{and} \tag{12}$$

$$\hat{c}_{\mathrm{i}} = R_{w,\mathrm{i}}[u,v] \cdot \overline{W}_{\gamma,o_{\hat{c}}} \cdot o_{\hat{c}} \tag{13}$$

are obtained in the loop. Here, it should be mentioned that the weighting scaling constant $o_{\hat{c}} = 2^{16}$ was chosen for (11), but also other values are possible, respecting the DC part of the weighting function. For example, $o_{\hat{c}} = 2^{17}$ and $o_{\hat{c}} = 2^{18}$ are verified to work with weighting function from (2), while not having significant influence on the reconstruction quality in terms of PSNR.

Another adaption is needed when updating the model $G[u,v]$. While the original algorithm adjusts the expansion coefficient $\hat{c}$ by multiplying with FFT size $M \times N$, the fixed point algorithm renounces these operations, because of its scaled input signal. Due to this, (6) simplifies to

$$G^{(\nu)}[u^{(\nu)}, v^{(\nu)}] = G^{(\nu-1)}[u^{(\nu)}, v^{(\nu)}] + \hat{c}_{(u^{(\nu)}, v^{(\nu)})} \,. \tag{14}$$

After the iterative reconstruction has finished, the generated model $g[m,n]$ has to be transformed back into the time domain using the inverse FFT. Before one can fill the distorted areas $\mathcal{B}_i$ with the reconstructed values, it must be kept in mind that all pixels have to get back-scaled by FFT window size $M \times N$ in general. Due to this,

$$g[m,n] = \mathrm{IFFT}\{G[u,v]\} \cdot M \cdot N \tag{15}$$

results.

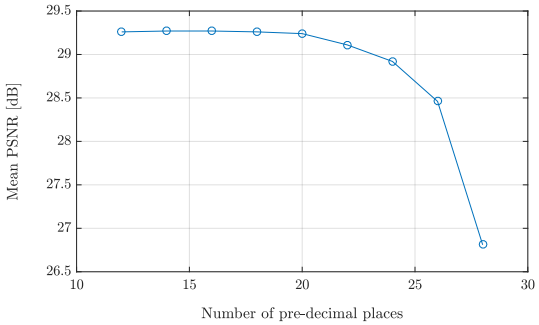


Fig. 3. Mean PSNR addicted to the number of pre-decimal digits for weighting function.

3.2. Weighting function

Using a data type with a limited value range does not fit for weighting function generation. Regarding (2), the exponent contains the FFT window height and width and the square operation is performed. So, it is obvious that big values emerge. Insertion of the maximal usual window size 64×64 for FSE in [9] leads to worst case value $Z_{\mathrm{WF}} = (m-(M-1)/2)^2 + (n-(N-1)/2)^2 = 1984.5$, which result in the use of $\log_2(1984.5) \geq 11$ bit. Moreover, signed integers shall be used, since overflows result in a sign discontinuity and are therefore easy to detect in the case of an error. This increases the necessary number of bits from 11 to 12. As shown in Fig. 3, it is advisable to choose some more pre-decimal digits. In the region of 12 to 20 digits, the reconstruction quality is constant, but decreases rapidly afterwards, because the precision is not sufficient anymore then. To have additional back-up for future algorithm adaptions, the mean value 16 of the flat region is chosen.

At this place, it should be mentioned that the programming language C does not provide the fixed-point functions for calculation the power or taking the square root. For a fast implementation, the power a^b is approximated by calculating

$$a^b \approx a^{\lfloor b \rfloor} = \prod_{1}^{\lfloor b \rfloor} a \,.$$

Furthermore, the square root was calculated using Heron's method [14], which provides a good approximation in a few steps only. For the fixed-point algorithm an upper limit of 6 iterations was chosen. A bigger number does not lead to a higher PSNR, but slows down the calculations.

4. EVALUATION AND RESULTS

For comparing the novel fixed-point and the state-of-the-art floating-point FSE algorithm different platforms were chosen. On the investigated systems, the implementation was compiled with GCC 5.3.1 running Ubuntu 16.04 LTS with Linux kernel 4.4.0. Only, the examined "Zybo Zynq" board uses GCC 4.6.3 running Ubuntu 12.04 LTS with Linux kernel 3.12.0, because of the underlying ARM architecture. First system named "Desktop" contains the state-of-the-art processor i7-6700K @ 4.0 GHz equipped with 16 GB RAM. Second one named "Notebook" encloses an i7-6700HQ @ 2.6 GHz and 8 GB RAM. Third system "Netbook" is based on a Pentium N3540 @ 2.16 GHz processor and provides 4 GB RAM. Finally, "Zybo Zynq" contains an ARM Cortex-A9 @ 650 MHz processor and 512 MB RAM [15].

Table 1. Mean execution times per image of floating-point and fixed-point FSE for different platforms averaging over TECNICK image set.

	Floating-Point	Fixed-Point	Saving
Desktop	1.94 s	1.76 s	9.28 %
Notebook	2.30 s	2.11 s	8.26 %
Netbook	9.62 s	6.41 s	33.37 %
Zybo Zynq	68.65 s	40.88 s	40.45 %

Concerning FSE, extrapolation area $\mathcal{L}$ and FFT window size was set to 32×32 and block b to 16×16. Decay $\hat{\rho}$ was set to 0.8 and weighting of already concealed areas δ to 0.2. For performing error concealment, block losses of size 16×16 were inserted into the images of the TECNICK dataset [16].

At first, it can be mentioned that there are no significant differences in the reconstruction quality for the novel approach. For all tested systems, a mean PSNR of 29.27 dB for fixed-point and 29.26 dB for floating-point systems was measured using TECNICK image set. Regarding every single image, the differences in terms of PSNR are in range of second decimal digit only.

However, the proposed fixed-point algorithm is faster than the state-of-the-art FSE. While the savings are about 9.28 % for the high-end "Desktop" system, one can obtain much higher benefits for cheaper hardware. Examining "Zybo Zynq" an execution time decrease of 40.45 % can be mentioned (see Tab. 1). This is due to the fact, that ARM architectures have the disadvantage of slower Floating-Point Units, while being more energy efficient compared to Intel processors [17].

5. CONCLUSION

In this paper, we presented a novel scaled fixed-point FSE. The concealment of errors in image data using the new approach achieves an average acceleration of 22.84 % and a maximum speedup of 40.45 % at the same reconstruction quality. This is obtained by exploiting the fact that state-of-the-art computers can perform integer calculations faster than floating-point ones. To deal with limitations of fixed-point arithmetics, e.g., reduced value range, the scaled FSE was introduced. For the future, it is now possible to bring the FSE on further platforms, e.g, FPGAs, which benefit from the use of fixed-point arithmetic. Moreover, we want to investigate the combination of the novel algorithm with the approaches [6] and [7] and the suitability for video signals.

6. REFERENCES

[1] T. Stockhammer and M. M. Hannuksela, "H.264/AVC video for wireless transmission," *IEEE Wireless Communications*, vol. 12, no. 4, pp. 6–13, Aug 2005.

[2] R. T. Pushpalwar and S. H. Bhandari, "Image inpainting approaches - a review," in *2016 IEEE 6th International Conference on Advanced Computing (IACC)*, Feb 2016, pp. 340–345.

[3] B. A. Olshausen and D. J. Fieldt, "Sparse coding with an overcomplete basis set: a strategy employed by v1," *Vision Research*, vol. 37, pp. 3311–3325, 1997.

[4] E. Candes, N. Braun, and M. Wakin, "Sparse signal and image recovery from compressive samples," in *2007 4th IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, April 2007, pp. 976–979.

[5] A. Kaup, K. Meisinger, and T. Aach, "Frequency selective signal extrapolation with applications to error concealment in image communication," *AEUE - International Journal of Electronics and Communications*, vol. 59, pp. 147–156, 2005.

[6] W. Schnurrer, M. Jonscher, J. Seiler, T. Richter, M. Bätz, and A. Kaup, "Centroid adapted frequency selective extrapolation for reconstruction of lost image areas," in *IEEE International Conference on Visual Communications and Image Processing (VCIP)*, Singapore, December 2015, pp. 1–4.

[7] M. Bätz, W. Schnurrer, J. Koloda, A. Eichenseer, and A. Kaup, "Joint shape and centroid adaptive frequency selective extrapolation for the reconstruction of arbitrarily shaped loss areas," in *Picture Coding Symposium*, Nuremberg, Germany, December 2016.

[8] X. Ma, W. A. Najjar, and A. K. Roy-Chowdhury, "Evaluation and acceleration of high-throughput fixed-point object detection on fpgas," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 6, pp. 1051–1062, June 2015.

[9] J. Seiler and A. Kaup, "Complex-valued frequency selective extrapolation for fast image and video signal extrapolation," *IEEE Signal Processing Letters*, vol. 17, no. 11, pp. 949 – 952, November 2010.

[10] ——, "Fast orthogonality deficiency compensation for improved frequency selective image extrapolation," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Las Vegas, Nevada, March 2008, pp. 781–784.

[11] E. L. Oberstar, *Fixed-Point Representation & Fractional Math*. Oberstar Consulting, August 2007.

[12] *6th Generation Intel Processor Datasheet for S-Platforms*, Intel, May 2016, datasheet, Volume 1 of 2.

[13] M. Borgerding, "Kiss FFT," 2013, v 1.3. [Online]. Available: https://sourceforge.net/projects/kissfft

[14] O. Kosheleva, "Babylonian method of computing the square root: Justifications based on fuzzy techniques and on computational complexity," in *NAFIPS 2009 - 2009 Annual Meeting of the North American Fuzzy Information Processing Society*, June 2009, pp. 1–6.

[15] L. H. Crockett, R. A. Elliot, M. A. Enderwitz, and R. W. Stewart, *The Zynq Book: Embedded Processing with the Arm Cortex-A9 on the Xilinx Zynq-7000 All Programmable Soc*. UK: Strathclyde Academic Media, 2014.

[16] N. Asuni and A. Giachetti, "Testimages: a large-scale archive for testing visual devices and basic image processing algorithms," *STAG - Smart Tools & Apps for Graphics Conference*, 2014.

[17] V. Nikolskiy and V. Stegailov, "Floating-point performance of arm cores and their efficiency in classical molecular dynamics," *Journal of Physics: Conference Series*, vol. 681, no. 1, p. 012049, 2016.