Fontes *et al. Journal of the Brazilian Computer Society* 2014, **20**:18
http://www.journal-bcs.com/content/20/1/18

Journal of the Brazilian Computer Society
a SpringerOpen Journal

RESEARCH Open Access

On the parallel efficiency and scalability of the correntropy coefficient for image analysis

Aluisio I Rêgo Fontes*, Samuel Xavier-de-Souza, Adrião D Dória Neto and Luiz Felipe de Queiroz Silveira

Abstract

Background: Similarity measures have application in many scenarios of digital image processing. The correntropy is a robust and relatively new similarity measure that recently has been employed in various engineering applications. Despite other competitive characteristics, its computational cost is relatively high and may impose hard-to-cope time restrictions for high-dimensional applications, including image analysis and computer vision.

Methods: We propose a parallelization strategy for calculating the correntropy on multi-core architectures that may turn the use of this metric viable in such applications. We provide an analysis of its parallel efficiency and scalability.

Results: The simulation results were obtained on a shared memory system with 24 processing cores for input images of different dimensions. We performed simulations of various scenarios with images of different sizes. The aim was to analyze the parallel and serial fraction of the computation of the correntropy coefficient and the influence of these fractions in its speedup and efficiency.

Conclusions: The results indicate that correntropy has a large potential as a metric for image analysis in the multi-core era due to its high parallel efficiency and scalability.

Keywords: Correntropy; Similarity measures; Multi-core architecture; Parallel scalability; Parallel efficiency

Background

Introduction

Similarity measures have been widely used in computer vision and digital image processing [1]. Principe et al. [2] developed a similarity measure sensitive to high-order statistics and temporal structures of random processes [3], called correntropy, that was inspired by the work of Parzen [4]. The presence of high-order statistics and temporal structures in the measure makes it suitable for many different applications.

In recent years, the concept of correntropy has been successfully applied to the solution of many problems related to digital image processing, such as automatic face recognition [5,6], facial recognition with occlusion [7], image recognition using MACE filter (correntropy minimum average correlation energy (CMACE) [8], and classification of sidescan sonar imagery [9]. Nevertheless, depending on the input size, the correntropy can demand a high computational cost. In order to exemplify, the output of the CMACE filter proposed in [8] is obtained by computing the product of two matrices, which depends on the image size and the number of training images which can restrict its application. In [10] and in [11], in order to alleviate the computational complexity, the authors proposed a method to estimate the correntropy using the fast Gauss transform (FGT) and the incomplete Cholesky decomposition (ICD), respectively. The FGT has been applied to many areas including astrophysics, kernel density estimation, and machine learning algorithms decreasing the computation from $O(NM)$ to $O(N + M)$, where N is the number of samples (sources) and M is the number of points where the evaluation is required [12]. Moreover, the correntropy can be expressed in terms of positive definite matrices, and ICD can be directly applied to reduce the computational burden.

In addition, there are some methods to estimate the correntropy based on the additive form of half-quadratic minimization and nonconvex relaxation [13-15]. There is, therefore, a quest to improve the computational cost of the

*Correspondence: aluisio@dca.ufrn.br
[1]Department of Computer Engineering and Automation, Lagoa Nova, 59078-970 Natal, Brazil

correntropy that may, or may not, involve approximations of the measure.

Another way to allow the use of the correntropy measure in more demanding applications is to find a strategy to make its computation scalable and efficient for a given architecture. This way, the scaling of the architecture will guarantee a faster computation of the measure. Scalability was a common characteristic of all algorithms before the multi-core era [16,17] for conventional architectures: any scaling or advance in the computer architecture technology generated a gain in the speed of computation. In the advent of the multi-core era, the scaling of the technology - provided by the increasing number of transistors per die - is realized by increasing the number of processing cores per chip and not by faster cores. The computational flow has now multiple paths that need to be subjected to synchronization, communication, and load balancing. The result is that scalability of algorithms in the multi-core era is no longer a common attribute of all algorithms. Now, scalability arises from intrinsic characteristics of the computation and from the programmer's ability to cope with synchronization, communication, and load balancing among the multiple computational flows. Therefore, computational work flows, such as the calculation of the correntropy, need to be evaluated to investigate what are the benefits or the drawbacks of employing it in this new computational era.

In single-core processors, the calculation of correntropy demands a large amount of time, mainly due to its $O(n^2)$ quadratic computational complexity. Depending on the amount of data, the response time can limit or even prevent the use of the metric in applications with higher input dimensions, which unfortunately includes the majority of applications involving computer vision and image analysis.

In this paper, we propose a strategy to calculate the correntropy metric in the form of a parallel algorithm for multi-core architectures. We provide an analysis of its parallel efficiency and scalability. The simulation results were obtained on a shared memory system with 24 processing cores for input images of different dimensions. The results show that correntropy has a large potential as a metric for image analysis in the multi-core era due to its high parallel efficiency and scalability.

The rest of the paper is organized as follows. Section 'Correntropy' has a brief description of the correntropy metric. In Section 'Parallel efficiency and scalability', we present the parallel performance measures used in the analysis of parallel efficiency and scalability. The implementation of parallel correntropy is presented in Section 'Parallel implementation of the correntropy coefficient', and the results are described in Section 'Results and discussion', while Section 'Conclusions' is intended to the conclusions of this work.

Methods

Correntropy

In [18], the use of metrics based on information theory in problems of machine learning was introduced. Originally, these problems were solved using measurements of the mean square error. The new methodology became known as information-theoretic learning (ITL). The idea of ITL is to use metrics based on nonparametric estimates of Renyi's quadratic entropy as cost functions for the design of adaptive systems. In the past few years, this concept has been successfully applied in the solution of various engineering problems, e.g., automatic modulation classification [19], prediction method for network traffic [20], adaptive filtering [21], and face recognition [22].

Renyi's definition of entropy is a generalization of Shannon's entropy, given by [12]

$$H_\alpha(X) = \frac{1}{1-\alpha}\log\left(\sum_{k=1}^{m} p_k^\alpha\right), \tag{1}$$

where p_k is defined from the probability distribution of random variable X and α is a parameter that specifies the order of the Renyi's entropy. When the probability distribution of X is estimated from a finite set of n measured data, $(x_i)_{i=1}^n$, by a Gaussian kernel, $G_\sigma(\cdot)$, with standard deviation σ, the Renyi's quadratic entropy (for $\alpha = 2$) can be estimated as [4]

$$\hat{H}_2(X) = -\log\left[\frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} G_{\sigma\sqrt{2}}(x_j - x_i)\right], \tag{2}$$

where,

$$G_\sigma(x_j - x_i) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left[-\frac{(x_j - x_i)^2}{2\sigma^2}\right]. \tag{3}$$

The argument of Renyi's quadratic entropy in Equation (2) is called information potential (IP) of the r.v. X, given by [12]

$$\hat{V}_\sigma(X) = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} G_{\sigma\sqrt{2}}(x_j - x_i). \tag{4}$$

The information potential can be generalized to two random variables, giving rise to the cross-information potential (CIP), defined by

$$\hat{V}_\sigma(X, Y) = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} G_{\sigma\sqrt{2}}(x_i - y_j). \tag{5}$$

For independent random variables, it can be demonstrated that [12]

$$\frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} G_{\sigma\sqrt{2}}(x_i - y_j) \approx \frac{1}{n}\sum_{i=1}^{n} G_{\sigma\sqrt{2}}(x_i - y_i), \tag{6}$$

which is an approximation of CIP with algorithmic complexity $O(n)$.

From the viewpoint of kernel methods, the right-hand side of Equation (6) can be seen as a sample estimator of a generalized similarity measure between two arbitrary scalar random variables X and Y, called *cross-correntropy* and defined by [2]

$$\begin{aligned} v_\sigma(X,Y) &= E_{XY}[\,G_\sigma(X-Y)] \\ &= \iint G_\sigma(X-Y)p_{X,Y}(x,y)dxdy, \end{aligned} \tag{7}$$

where $E[\,\cdot]$ is the expectation operator and σ is the kernel size.

By estimating the joint probability density function (pdf) of Equation (7) by the Parzen method from a finite number of data $\{(x_i, y_i)\}_{i=1}^n$, the right-hand side of Equation (6) is obtained

$$\hat{v}_\sigma(X,Y) = \frac{1}{n}\sum_{i=1}^{n} G_{\sigma\sqrt{2}}(x_i - y_i). \tag{8}$$

It is interesting to notice the link between cross-correntropy estimators and information-theoretic estimators, when the variables X and Y are independent.

By applying the Taylor series expansion on the Gaussian function in Equation (7) and assuming that all the moments of the joint pdf are finite, Equation (7) yields

$$v_\sigma(X,Y) = \frac{1}{\sqrt{2\pi}\sigma}\sum_{k=0}^{\infty}\frac{(-1)^k}{2^k\sigma^{2k}k!}E[\,(X-Y)^{2k}]\,. \tag{9}$$

This expression affirms that the cross-correntropy is sensitive to the sum of second-order moment and higher-order moment of the difference variable. Thus, cross-correntropy is considered a generalization of the correlation [2], and as well as correlation, it can be used as a measure of similarity between random variables. In fact, since cross-correntropy is sensitive to the sum of all even moments of the random variables, in many applications, it may quantify the relationships between these variables better than correlation, as in non-Gaussian and nonlinear problems [2].

The kernel size in Equation (9) appears as a parameter that weighs the second-order moment and the higher-order moment. For sufficiently, large values of σ, the second-order moment dominates and the measure approaches correlation.

Due to nonlinear transformations produced by the Gaussian kernel, cross-correntropy has no guarantee of zero mean, even when the input data are centered at zero. The definition of centered cross-correntropy overcomes this limitation [12]

$$\hat{u}_\sigma(X,Y) = \frac{1}{n}\sum_{i=1}^{n}G_{\sigma\sqrt{2}}(x_i - y_i) - \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}G_{\sigma\sqrt{2}}(x_i - y_j). \tag{10}$$

We can notice that the centering term in Equation (10) is numerically equal to the estimator of the cross-information potential in Equation (5). Thus, from the Equation (6), centered cross-correntropy reduces to zero if X and Y are independent random variables.

In applications involving signals with unknown amplitudes, conducting some type of normalization can be required. To avoid this process, Xu et al. [3] presented a new similarity measure called *correntropy coefficient*, estimated by

$$\hat{\eta} = \frac{\hat{u}_\sigma(X,Y)}{\sqrt{\hat{u}_\sigma(X,X)}\sqrt{\hat{u}_\sigma(Y,Y)}} \tag{11}$$

The correntropy coefficient $\hat{\eta}$ can be considered a generalization of the known correlation coefficient. It can be verified that its value reduces to zero if the two random variables X and Y are statistically independent, and its absolute value is close to one as they become statistically related.

In this paper, we propose a parallel algorithm for calculating the correntropy coefficient. We analyze its parallel efficiency and scalability toward very large input dimensions. For each image size, we constructed two random images and measured the similarity using the proposed parallel correntropy coefficient. We choose random images over other types of images because we are not actually interested in the similarity measures themselves but rather in the computational performance of these measures.

Parallel efficiency and scalability

In parallel computing, speedup is defined as the ratio between the sequential execution time T_s and the parallel execution time T_p, given by

$$S = \frac{T_s}{T_p}. \tag{12}$$

It indicates how many times the parallel algorithm is faster than sequential one.

A linear or ideal speedup is obtained for example when by doubling the number of processors also doubles the speed of processing [23]. Thus, the goal for parallel algorithms is trying to achieve linear speedup. However, according to Amdahl's law [24], the parallel speedup is limited by the sequential fraction of the execution time. The sequential fraction F_s of an algorithm executing in a given architecture with p processors can be estimated by the following expression

$$F_s = \frac{pT_p - T_s}{pT_s - T_s}. \tag{13}$$

Parallel efficiency is a value, typically between zero and one, that expresses the percentage of the speedup achieved by the algorithm compared to the linear speedup. It represents the percentage of processing power that is actually being used to perform the calculation. The complement of the efficiency indicates how much effort is being wasted due to parallelization overhead. Efficiency E is expressed by

$$E = \frac{T_s}{pT_p} \tag{14}$$

where p is the number of processing elements used in the parallel execution. Note that, if $T_p = T_s/p$, we would have all processors doing useful work, leading to an efficiency of 100%.

When analyzing the scalability of a parallel algorithm, it is important to observe the values of the sequential fraction and the parallel efficiency when the number of processor and the problem size scales. Generally, we would expect that the sequential fraction do not scale together with these numbers. As for the parallel efficiency, since its value is inversely proportional to p, commonly, we observe a decreasing value with an increase in the number of processing elements. For this reason, the scalability of a parallel algorithm is generally associated, especially for larger problems, with the scaling of the problem size [25].

In this work, we study the speedup, the efficiency, and the scalability of a parallel algorithm for the calculation of correntropy. Also, a study of the sequential fraction of the algorithm is performed in order to provide insights on how much time is spent with communication, synchronization, and load balancing.

Parallel implementation of the correntropy coefficient

The correntropy coefficient was implemented in parallel using the OpenMP framework [26]. In order to facilitate the parallelization, we organized the sequence of operations for the calculation of the correntropy coefficient.

Substituting the definitions of the cross correntropy Equation (8) and the cross information potential Equation (5) into the correntropy coefficient Equation (12), the following expression is obtained

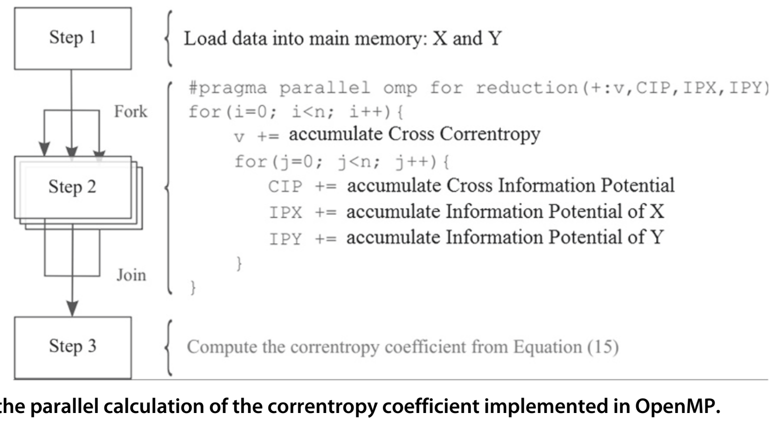
$$\hat{\eta} = \frac{\hat{v}_\sigma(X,Y) - \hat{V}_\sigma(X,Y)}{\sqrt{\left[\hat{v}_\sigma(X,X) - \hat{V}_\sigma(X,X)\right]\left[\hat{v}_\sigma(Y,Y) - \hat{V}_\sigma(Y,Y)\right]}}, \tag{15}$$

where $\hat{v}_\sigma(X,X) = \hat{v}_\sigma(Y,Y) = \frac{1}{n}\sum_{i=1}^{n} G_{\sigma\sqrt{2}}(0)$ is a constant that depends only on the kernel size σ and is denoted herein by κ_σ, and $\hat{V}_\sigma(X,X)$ and $\hat{V}_\sigma(Y,Y)$ are the IP estimators for X and Y, respectively. Thus, the correntropy coefficient can be obtained by

$$\hat{\eta} = \frac{\hat{v}_\sigma(X,Y) - \hat{V}_\sigma(X,Y)}{\sqrt{\left[\kappa_\sigma - \hat{V}_\sigma(X)\right]\left[\kappa_\sigma - \hat{V}_\sigma(Y)\right]}}, \tag{16}$$

where $\hat{v}_\sigma(X,Y)$ is calculated by a single summation in the index i, while $\hat{V}_\sigma(X,Y)$, $\hat{V}_\sigma(X)$, and $\hat{V}_\sigma(Y)$ are calculated by double summations in the indexes i and j.

Figure 1 illustrates the pseudo-code of the parallelized calculation of this metric. The first step of the algorithm is responsible for loading the characteristics of the images and is not considered for execution time measurements. In the second step, the data are divided among the threads for computation. Note that, only two loops are used to compute all summations. While the inner and the outer loops are used to compute all information potentials, $\hat{V}_\sigma(X,Y)$, $\hat{V}_\sigma(X)$, and $\hat{V}_\sigma(Y)$, the cross correntropy, $\hat{v}_\sigma(X,Y)$, is computed only in the outer loop. In the last



```
Load data into main memory: X and Y

#pragma parallel omp for reduction(+:v,CIP,IPX,IPY)
for(i=0; i<n; i++){
    v += accumulate Cross Correntropy
    for(j=0; j<n; j++){
        CIP += accumulate Cross Information Potential
        IPX += accumulate Information Potential of X
        IPY += accumulate Information Potential of Y
    }
}

Compute the correntropy coefficient from Equation (15)
```

Figure 1 Pseudo-code of the parallel calculation of the correntropy coefficient implemented in OpenMP.

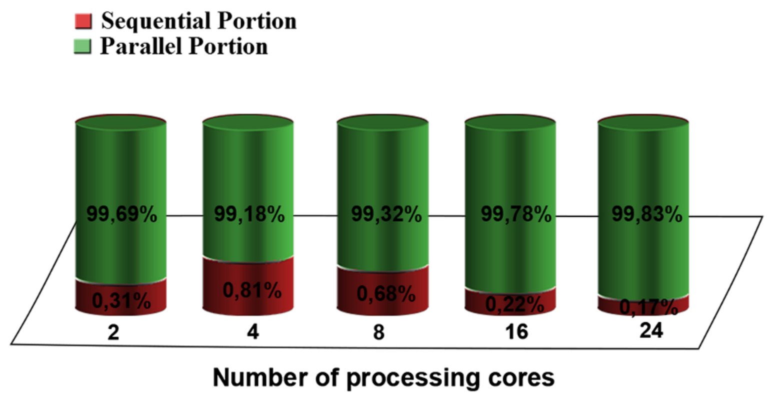


Figure 2 Analysis of the parallel and serial fraction of the proposed parallel algorithm. To calculate the correntropy coefficient for images with dimension 200 × 200.

step of the algorithm, the main thread is responsible to calculate the final value of the correntropy coefficient.

The serial algorithm for the calculation of the correntropy coefficient has an arithmetic complexity of $O(n^2)$. In the proposed method, each one of the p processors runs the same program but with distinct data, yielding an arithmetic complexity of the correntropy coefficient to $O(\frac{n^2}{p})$.

Another important performance measure is the space complexity defined as memory required by the algorithm. The memory requirement of the proposed method is $2k(n \times m)$, where n and m are image dimensions and k is the number of bytes needed to stored the data type in the memory.

Results and discussion

The experiments were performed on a system with two AMD Opteron 6172, each with 12 cores running at 2.1 GHz. We performed simulations of various scenarios with images of different sizes. The aim was to analyze the parallel and serial fraction of the computation of the correntropy coefficient and the influence of these fractions in its speedup and efficiency. For the calculation of the correntropy coefficient, we used a kernel size σ equal to $\frac{1}{\sqrt{2\pi}}$ for all simulations. In order to evaluate the performance of this measure for image analysis applications, we selected image sizes of 50 × 50, 100 × 100, 200 × 200, and 400 × 400 because they may well represent a large range of sizes used in filtering and convolution of images. For each image size, we constructed two random images and measured the similarity using the proposed parallel correntropy coefficient. Each value of execution time used in the analysis was obtained through an arithmetic average over five measurements for each image size.

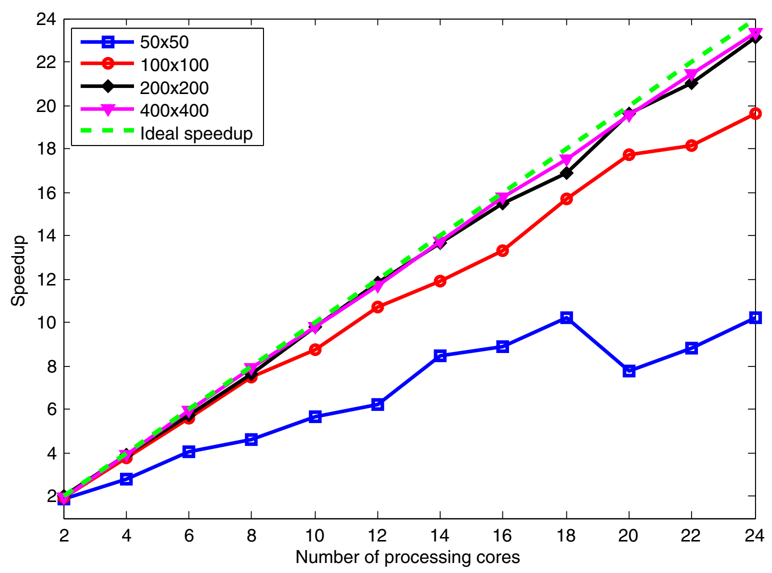


Figure 3 Speedups of the proposed algorithm for images with sizes 50 × 50, 100 × 100, 200 × 200, and 400 × 400.

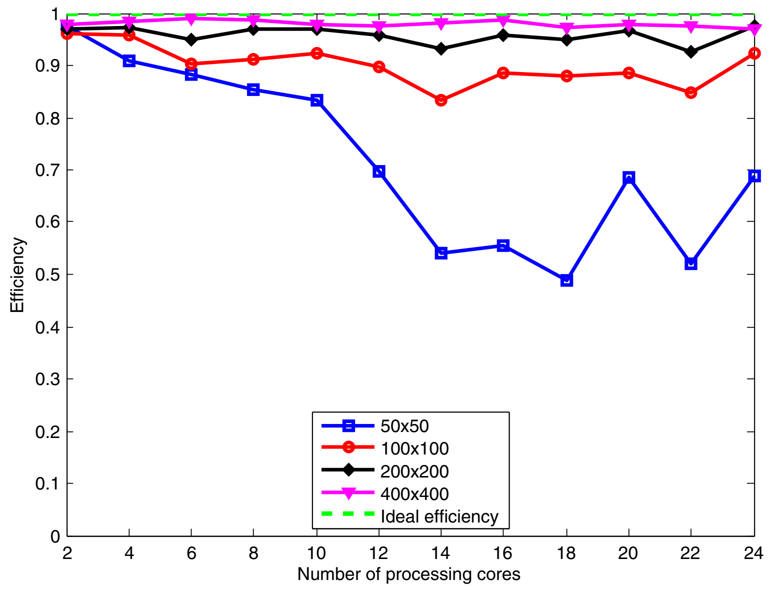


Figure 4 Results of efficiency for different image sizes 50 × 50, 100 × 100, 200 × 200, and 400 × 400.

Sequential fraction

Generally, every parallel algorithm has a fraction of code that needs to be executed sequentially. The complement of this fraction can be concurrently executed. The increase in processing speed obtained with the parallelization of the algorithm is limited by this sequential fraction. Thus, from Equation (13), we analyzed the serial fraction of the proposed algorithm to discover its potential for parallelization with images of size 200×200. The results are shown in Figure 2.

From the results shown in Figure 2, we observed that the sequential fraction of the algorithm is always much smaller than the parallel fraction. It increases from two to four processors but decreases asymptotically for higher numbers of processors. The decreasing of the parallel fraction of the proposed algorithm with the growing number of processors indicates a potential for parallelization.

Speedup

With the objective to evaluate the speedup of the proposed parallelization for the calculation of the correntropy coefficient for each image set, we conducted simulations for a varying number of processing elements. The obtained results are presented in Figure 3.

We observe from Figure 3 that the performance of the proposed algorithm with up to eight processing cores reached nearly ideal speedup for the three largest image sizes. From ten processing cores and up, the computation for smaller images, 50×50 and 100×100, had a reduced speedup, probably due to the overhead of creation of threads only to process a limited amount of data. On the other hand, for images larger than 200×200, the speedups obtained by increasing the amount of processing cores were ever increasing. However, for 50×50 image size, we were not able to get linear speedup.

Parallel efficiency

The analysis of parallel efficiency has the objective of verifying how much processing is actually being used in relevant computation and how much is spent on the parallelization effort itself, which is the extra time spent on communication, synchronization, and load balancing. Figure 4 illustrates the results of efficiency for different image sizes.

Observe in Figure 4 that as the image sizes grow so does the efficiency. Even for large numbers of processing cores, where the efficiency is expected to drop, the value sustains for larger image sizes. These results are evidence that the proposed parallel implementation of the correntropy coefficient has a very good scalability w.r.t. the scalings of the multi-core era.

Conclusions

We presented a parallel algorithm for calculating the correntropy coefficient and analyzed its parallel efficiency and scalability for image analysis in the context of the multi-core era. The results show that the proposed implementation is scalable and efficient. Specially for the larger image sizes that we analyzed, the efficiency and scalability of the algorithm presented outstanding results. Another important contribution of the results presented in the paper, besides the methodology for devising an efficient and scalable implementation for correntropy coefficient, is the ratification of the correntropy coefficient as a robust measure of similarity. The metric can be safely employed in parallel image analysis or computer vision applications without threatening the overall performance of the application. The designer of such applications can use the measure of correntropy as a robust, efficient, and scalable building block, needing to worry only about the performance of the rest of the application.

Competing interests
The authors declare that they have no competing interests.

Authors' contributions
All authors have contributed to the different methodological and experimental aspects of the research. All authors read and approved the final manuscript.

Authors' information
Aluisio I.R. Fontes received the Computer Engineering at UFRN/Brazil in 2010 and his MSc in Computer Engineering from UFRN in 2012. He is currently a doctoral student in Electrical Engineering at Federal University of Rio Grande do Norte (UFRN). His current interests include cognitive radio, information theoretic learning, telecommunication system, parallel software scalability, and parallel algorithms.
Samuel Xavier-de-Souza received the Computer Engineering degree at UFRN, in 2000 and his Ph.D. in Electrical Engineering from Katholieke Universiteit Leuven (KULeuven), Belgium, in 2007. He worked with the Interuniversity MicroElectronics Center (IMEC), as a software engineer and with the Flemish Supercomputer Centre in Belgium as a research consultant and in since 2009 is a professor at the Department of Computer Engineering and Automation of UFRN. His research interests are in parallel software efficiency, parallel software scalability, parallel algorithms, and simulation of complex systems.
Adrião D.D. Net received the B.S. degree in Electrical Engineering from Universidade Federal do Rio Grande do Norte (UFRN), Natal, Brazil, in 1978; the M.S. degree in Information Processing and Electromagnetic Theory from University of Campinas (UNICAMP), Barão Geraldo, Brazil, in 1982; and the Ph.D. degree in Electronics from the Ecole Nationale Superieured'Electrotechnique, d'Electronique, d'Informatique et d'Hydraulique de Toulouse (ENSEEIHT), France, in 1989. Currently, he is a professor at the Department of Computing and Automation Engineering, UFRN. His research is related to computational intelligence and digital signal processing.
Luiz F. Q. Silveira received his bachelor degree in Electrical Engineering, from Universidade Federal da Paraíba (UFPB), Brazil, 2000; his master degree (2002); and Ph.D. (2006) in Electrical Engineering, from Universidade Federal de Campina Grande (UFCG), Brazil. Since 2010, he is a professor at the Department of Computer Engineering and Automation, Universidade Federal do Rio Grande do Norte, Brazil. In 2006, he was also a professor in Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte. He has experience in the electric engineering area, with emphasis in telecommunication systems, acting mainly in the following subjects: channel coding, digital signal processing, and cognitive radio.

Acknowledgements
The author Aluisio Fontes is supported by the Committee for the Advancement of University Academic Staff - CAPES.

Received: 20 March 2014 Accepted: 28 September 2014
Published online: 16 October 2014

References

1. Chalom E, Asa E, Biton E (2013) Measuring image similarity: an overview of some useful applications comparing images using a pattern. IEEE Instrum Meas Mag 16:24-28
2. Santamaría I, Member S, Pokharel PP, Member S, Principe JC (2006) Generalized correlation function: definition, properties, and application to blind equalization. IEEE Trans Signal Process 54:2187-2197
3. Xu J-W, Bakardjian H, Cichocki A, Principe JC (2008) A new nonlinear similarity measure for multichannel signals. Neural Netw 21:222-231
4. Parzen E (1962) On estimation of a probability density function and mode. Ann Math Stat 1:1065-1076
5. He R, Zheng W-S, Hu B-G, Member S (2013) Two-stage nonnegative sparse representation for large-scale face recognition. IEEE Trans Neural Netw Learn Syst 24:35-46
6. He R, Zheng W-S, Hu B-G (2011) Maximum correntropy criterion for robust face recognition. IEEE Trans Pattern Anal Mach Intell 33:1561-1576
7. Li X-X, Dai D-Q, Zhang X-F, Ren C-X (2013) Structured sparse error coding for face recognition with occlusion. IEEE Trans Image Process 22:1889-900
8. Jeong K-H, Liu W, Han S, Hasanbelliu E, Principe JC (2009) The correntropy MACE filter. Pattern Recognit 42:871-885
9. Hasanbelliu E, Principe JC, Slatton C (2009) Correntropy based matched filtering for classification in sidescan sonar imagery. IEEE Int Conf 1:2757-2761
10. Jeong K-H, Han S, Principe JC (2007) The fast correntropy mace filter. IEEE Speech Signal Process 2:613-615
11. Seth S, Principe JC (2009) On speeding up computation in information theoretic learning. Int Joint Conf Neural Netw:2883-2887
12. Principe JC (2010) Information theoretic learning: Rényi's entropy and kernel perspectives. Springer. 2010
13. Yang Y, Feng Y, Suykens JA (2014) Robust recovery of corrupted low-rank matrix by implicit regularizers. IEEE Trans Pattern Anal Mach Intell 36(4):770–783
14. He R, Zheng W, Tan T, Sun Z (2014) Half-quadratic-based iterative minimization for robust sparse representation. IEEE Trans Anal Mach Intell 36:261-275
15. Yang Y, Feng Y, Suykens JA (2013) A nonconvex relaxation approach to robust matrix completion. IJCAI'13 Proceedings of the Twenty-Third international joint conference on Artificial Intelligence, vol. 11, pp 1764-1770
16. Hill MD, Marty MR (2008) Amdahl's law in the multicore era. IEEE Comput Soc 41:33-38
17. Sun X-H, Chen Y, Byna S (2008) Scalable computing in the multicore era. Proceedings of the International Symposium on Parallel Architectures, Algorithms and Programming.
18. Principe JC, Xu D (1999) An introduction to information theoretic learning. IJCNN Neural Networks 3:1783-1787
19. Fontes AI, Martins A d M, Silveira LF, Principe JC (2014) Performance evaluation of the correntropy coefficient in automatic modulation classification. Expert Syst Appl 21:122-131
20. Qu H, Ma W, Zhao J, Wang T (2013) Prediction method for network traffic based on maximum correntropy criterion. IEEE Chin Commun 10:134-145
21. Chen B, Xing L, Liang J, Zheng N, Principe JC (2014) Steady-state mean-square error analysis for adaptive filtering under the maximum correntropy criterion. IEEE Signal Process Soc 21:880-884
22. Li X-X, Dai D-Q, Zhang X-F, Ren C-X (2013) Structured sparse error coding for face recognition with occlusion. IEEE Trans Image Process 22:1889-1900
23. Petersen WP, Arbenz P (2003) Introduction to parallel computing. Hardcover
24. Amdahl G (2007) Validity of the single processor approach to achieving large scale computing capabilities. IEEE Solid-State Circuits Soc Newslett 12:483-485
25. Gustafson JL (1988) Reevaluating Amdahl's law. Commun ACM 31:532-533
26. Chandra R (2001) Parallel Programming in OpenMP. Hardcover

doi:10.1186/s13173-014-0018-4
Cite this article as: Fontes *et al.*: **On the parallel efficiency and scalability of the correntropy coefficient for image analysis.** *Journal of the Brazilian Computer Society* 2014 **20**:18.