

 Open access • Journal Article • DOI:10.1007/S11042-014-2195-8

An improved lossless image compression based arithmetic coding using mixture of non-parametric distributions — [Source link](#)

Atef Masmoudi, William Puech, Afif Masmoudi

Institutions: University of Montpellier, University of Sfax

Published on: 01 Dec 2015 - Multimedia Tools and Applications (Springer US)

Topics: Context-adaptive binary arithmetic coding, Data compression, Arithmetic coding, Lossless compression and Context-adaptive variable-length coding

Related papers:

- [AN EFFICIENT ADAPTIVE ARITHMETIC CODING FOR BLOCK-BASED LOSSLESS IMAGE COMPRESSION USING MIXTURE MODELS](#) Atef MASMOUDI † Afif MASMOUDI William PUECH †
- [A new arithmetic coding model for a block-based lossless image compression based on exploiting inter-block correlation](#)
- [Satellite sequence image lossless compression method and system based on mixed entropy coding](#)
- [State-based dynamic multi-alphabet arithmetic coding for image compression](#)
- [Local Prediction for Lossless Image Compression.](#)

Share this paper:    

View more about this paper here: <https://typeset.io/papers/an-improved-lossless-image-compression-based-arithmetic-1ilzaubckx>

An improved lossless image compression based arithmetic coding using mixture of non-parametric distributions

Atef Masmoudi · William Puech · Afif Masmoudi

Received: 18 November 2013 / Revised: 1 July 2014 / Accepted: 14 July 2014
© Springer Science+Business Media New York 2014

Abstract In this paper, we propose a new approach for a block-based lossless image compression using finite mixture models and adaptive arithmetic coding. Conventional arithmetic encoders encode and decode images sample-by-sample in raster scan order. In addition, conventional arithmetic coding models provide the probability distribution for whole source symbols to be compressed or transmitted, including static and adaptive models. However, in the proposed scheme, an image is divided into non-overlapping blocks and then each block is encoded separately by using arithmetic coding. The proposed model provides a probability distribution for each block which is modeled by a mixture of non-parametric distributions by exploiting the high correlation between neighboring blocks. The Expectation-Maximization algorithm is used to find the maximum likelihood mixture parameters in order to maximize the arithmetic coding compression efficiency. The results of comparative experiments show that we provide significant improvements over the state-of-the-art lossless image compression standards and algorithms. In addition, experimental results show that the proposed compression algorithm beats JPEG-LS by 9.7 % when switching between pixel and prediction error domains.

Keywords Arithmetic coding · Lossless compression · Image compression · Finite mixture model · Expectation-maximization algorithm · Kullback-Leibler distance

A. Masmoudi
Sfax Preparatory Engineering Institute, University of Sfax, BP 1172-3018 Sfax, Tunisia
e-mail: atef.masmoudi@lirmm.fr

A. Masmoudi · W. Puech (✉)
LIRMM, UMR CNRS 5506, University of Montpellier II, 34392 Montpellier Cedex 05, France
e-mail: william.puech@lirmm.fr

A. Masmoudi
Laboratory of Statistics and Probability, Faculty of Sciences of Sfax, University of Sfax, Sfax, Tunisia
e-mail: afif.masmoudi@fss.rnu.tn

1 Introduction

Arithmetic coding (AC) is a very efficient technique for lossless data compression and it produces a rate which approaches the entropy [1, 21] of the encoded data. In addition, AC is widely used in the well known image and video compression algorithms such as JBIG, JBIG2, JPEG2000 and H.264/AVC. AC is a statistical coding technique which needs to work with a model that estimates the probability of each pixel at each iteration in the encoding and decoding processes. The model needs to find the probability distribution of symbols that maximizes the compression efficiency. The model can be static (SAC) [19], adaptive (AAC) or context-based adaptive (CAAC). When the adaptive model considers the probability of just the symbol, without any other additional information, it is called AAC order-0 (AAC-0) [20, 25].

Several image compression schemes based on AAC and CAAC have been proposed. Carpentieri [5] proposed to appropriately select, for each pixel position, one of a large number of possible dynamic probability distributions, and encoded the current pixel prediction error by using this distribution as a model for AC. The scheme presented by Carpentieri [5] is very slow and requires large amount of computations, especially for large images. Ye et al. [28] presented an experimental study on issues related to parametric probability modeling for entropy coding. The probability distributions used in [28] include the distributions most commonly used in image modeling, such as Laplacian distribution, Gaussian distribution, t-distribution and generalized Gaussian distribution. Matsuda et al. [15] proposed to encode image prediction errors using a kind of CAAC. They proposed to use a model that approaches the probability density of errors by a generalized Gaussian function which makes the encoding algorithm very slow. Nokutaba et al. [11] presented a model that estimates the probability density of each pixel error by a Laplacian distribution with zero mean. The compression ratios in [11] are on average 5 % higher than those by conventional AC. Zhao et al. [30] assumed that the source has a probability density function of Gaussian mixture models. In [17], Ninoo and Nguyen proposed a new framework for efficient entropy coding of data which can be represented by a parametric distribution model. The authors demonstrated that the results obtained from a parametric model based on geometric distribution are very impressive. In previous work [14], Masmoudi et al. presented an efficient lossless compression scheme for still images based on AC. The model used in this scheme is based on updating the pixel probabilities only after detecting the last occurrence of each pixel. In [3], the local average-based probability model was proposed, where the probabilities are determined using the immediate neighbors of the currently coded wavelet coefficient. In [23], Teuhola designed a probability model to map any source string to a unique integer. Zhang et al. [29] proposed to divide each source symbol into several segments and each segment is separately coded with an AAC. With this division, possible values of each segment are concentrated within a small range, which leads to a better fit of the adaptive model to the source symbols statistics. Recently, a block-based AAC, called BBAAC, which estimates the probability distribution of each block of pixels by minimizing the Kullback-Leibler distance between the exact probability distribution of that block and the probability distributions of its causal neighbor blocks has been proposed [13]. The compression results show significant improvements over conventional AC, reducing the bitrate by an average of 16.4 %. Estimation of the order of dependencies has been presented in [2]. More recently AC is also proposed for lossy image compression [27].

In this work, we define a new improved model-based AAC. The proposed model estimates the distribution of each block by a mixture of non-parametric distributions obtained from its causal neighboring blocks.

The rest of this paper is organized as follows. Section 2 presents the AC principle and Section 3 presents Finite Mixture Models (FMM). In Section 4, we detail our proposed method and present the mathematical proof. In Section 5, we present experimental results on the performance of the proposed approach. The conclusion is finally presented in Section 6.

2 Overview of arithmetic coding

In this section, we present the basic notions of AC [8, 12, 18, 25]. Initially, AC determines the symbols to be encoded, and the symbol probabilities. To estimate the symbol probabilities, AC works with a model which can be static or adaptive. Finally, the encoder performs the encoding algorithm and generates the corresponding bitstream.

In the SAC technique, the first step consists of calculating the exact frequency table for all symbols. However, AAC-0 [20] starts with the same frequency count for all symbols and updates the frequency of each symbol by incrementing its count just after it has been encoded, to approximate the observed frequencies. In static or fixed models, the exact frequencies must be transmitted before the bitstream is sent. However, in adaptive models no additional information is required to be sent to the decoder, which uses the same initial values and the same updating algorithm.

Let X be a message to compress composed of m samples $X = x_1, \dots, x_m$. Each sample x_i , with $i \in \{1, \dots, m\}$, takes value from an alphabet \mathcal{A} composed of n mutually distinct symbols. Let f denote the probability density function (PDF) of the symbols of the alphabet. Therefore, the number of bits required representing the sequence X , after applying SAC, denoted by N_{SAC} , is close to:

$$\begin{aligned}
 N_{SAC} &\approx -\log_2(L(X/f)) \\
 &\approx -\log_2\left(\prod_{i=1}^m f(x_i)\right) \\
 &\approx -\log_2\left(\prod_{k \in \mathcal{A}} f(k)^{c(k)}\right) \\
 &\approx -\sum_{k \in \mathcal{A}} c(k)\log_2(f(k)),
 \end{aligned}
 \tag{1}$$

where $L(X/f)$ corresponds to the likelihood of observed data X given a model f , and c is the true frequency count table with $m = \sum_{k \in \mathcal{A}} c(k)$. Hence, the average number of bits

per symbol is bounded by $\frac{N_{SAC}}{n}$, which corresponds to the entropy [21] of the data source given by:

$$H_f = -\sum_{k \in \mathcal{A}} f(k)\log_2(f(k)).
 \tag{2}$$

Now, if we propose to encode message X by an estimated PDF \hat{f} , then the number of bits needed to represent X , denoted by $N_{\hat{f}}$, is close to:

$$\begin{aligned}
 N_{\hat{f}} &\approx -\log_2(L(X/\hat{f})) \\
 &\approx -\sum_{k \in \mathcal{A}} c(k)\log_2(\hat{f}(k)),
 \end{aligned}
 \tag{3}$$

and the average number of bit per symbol is bounded by:

$$H_{\hat{f}} = - \sum_{i=1}^n f(x_i) \log_2(\hat{f}(x_i)). \quad (4)$$

Therefore, the number of additional bits when using distribution \hat{f} to encode message X is equal to:

$$\begin{aligned} H_{\hat{f}} - H_f &= - \sum_{i=1}^n f(x_i) \log_2(\hat{f}(x_i)) \\ &\quad + \sum_{i=1}^n f(x_i) \log_2(f(x_i)) \\ &= \sum_{i=1}^n f(x_i) \log_2\left(\frac{f(x_i)}{\hat{f}(x_i)}\right) \\ &= KL(f||\hat{f}), \end{aligned} \quad (5)$$

which corresponds to the Kullback-Leibler distance (KL) [10] of the exact probability table f with respect to the estimated one \hat{f} . Hence, a good model is one that minimizes the KL distance between the estimated and the exact PDF.

3 Finite mixture models

Finite Mixture Models (FMM) [16] provide powerful and flexible probabilistic models which have been used successfully in many applications such as density estimation, clustering and classification.

The mixture density $f(x)$ can be formulated as:

$$f(x, \psi) = \sum_{l=1}^K \pi_l f_l(x, \theta_l), \quad (6)$$

where the vector $\psi = (\pi_1, \dots, \pi_K, \theta_1, \dots, \theta_K)$ contains all parameters of the mixture such that $0 \leq \pi_l \leq 1$ and $\sum_{l=1}^K \pi_l = 1$, and each f_l is a density function parametrized by θ_l .

The mixture model likelihood function is the probability of observing the data given a model f with parameters vector ψ , and for m independent samples, $X = x_1, \dots, x_m$, it is given by:

$$L(X, \psi) = \prod_{i=1}^m f(x_i, \psi), \quad (7)$$

Maximizing the AC efficiency corresponds to maximizing the likelihood function $L(X, \psi)$. Thus, we wish to find ψ such that $L(X, \psi)$ is maximum. This is known as the maximum likelihood estimates (MLE) for ψ . In order to estimate ψ , it is typical to use the \log likelihood function, defined as:

$$l(X, \psi) = \ln(L(X, \psi)). \quad (8)$$

Note that the value of ψ which maximizes $L(X, \psi)$ also maximizes $l(X, \psi)$.

The FMM parameters can be estimated very efficiently through MLE using the iterative expectation maximization (EM) algorithm [6]. Each iteration of the EM-algorithm consists of two processes: the expectation or the E-step, and the maximization or the M-step. The EM

algorithm starts with initial guesses about the mixture parameters ψ . The E-step calculates the expected value of the $l(X, \psi)$ function using the current estimate for the parameters, and the M-step computes parameters maximizing the expected log-likelihood found in the E-step.

In this work, we limit our study to the case of mixture of non-parametric densities. In this case, the mixture parameters are only composed of the mixing proportions.

4 Proposed approach

A very important strategy to losslessly compresses images is to divide an image into small blocks of pixels and to separately encode each block with its exact PDF while the average entropy of blocks is lower than the entropy of the whole image. Table 1 shows the compression results, in bits per pixel (bpp), obtained when encoding a Lena image of size 512×512 with a SAC and without taking into account the additional information corresponding to the PDF of each block, which requires a very large amount of data. The result presented in Table 1 is very interesting and the coding rate is reduced from 5.63bpp to 3.38bpp when using a small block size. This result prompted us to define a probabilistic model for finding, for each block, the PDF that fits its exact density with a minimum of additional information, because if the probabilities of the samples in a block totally match those in the model, the compression efficiency will be optimal.

More specifically, we will focus our attention on defining a new model that efficiently estimates the PDF of each block of pixels with a minimum of additional information.

Therefore, at the beginning of lossless block encoding, we propose to find a mixture of non-parametric distributions that maximizes the likelihood function given that block based on the probability distributions of its neighboring blocks. Non-parametric models offer an important alternative to traditional parametric models for exploring large amounts of data without making specific distributional assumptions. AAC uses the probabilities generated by the founded mixture model and starts encoding symbols with these probabilities. As the coding process continues, AAC updates the symbol probabilities and they gradually converge to the real PDF of each block.

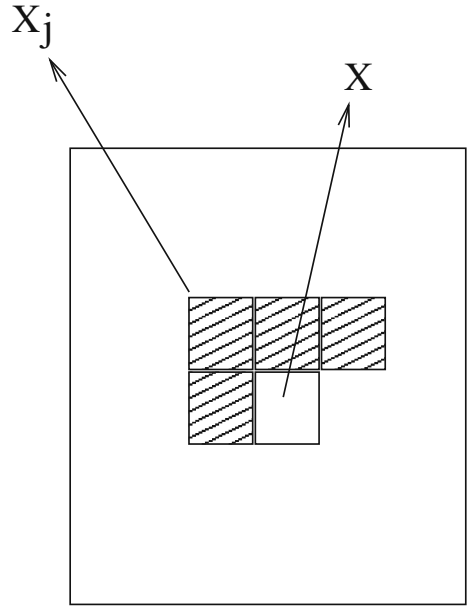
Let X be an image block to compress which is composed of m pixels $X = x_1, \dots, x_m$. Each pixel x_i , with $i \in \{1, \dots, m\}$, takes a value from an alphabet \mathcal{A} composed of n symbols. We propose to model the PDF of the block X , denoted by f_X , by a mixture of K probability distributions $\{f_{X_j}\}_{j=1}^K$, where f_{X_j} is a PDF calculated from X_j , the j th neighbor block, as illustrated in Fig. 1. Therefore, the probability that a certain pixel has a value k can be written as:

$$f_X(k) = \sum_{j=1}^K \pi_j f_{X_j}(k), k \in \mathcal{A}, \quad (9)$$

Table 1 Compression results for a Lena image (in bits/pixel) using SAC for each block

Block size			
32×32	16×16	8×8	4×4
5.63	5.00	4.26	3.38

Fig. 1 X and its neighboring blocks X_j



where π_j is the mixing proportion of the j th distribution, such that $0 \leq \pi_j \leq 1$ and $\sum_{j=1}^K \pi_j = 1$; K is the number of adjacent blocks used in the mixture model, and f_{X_j} is calculated as:

$$f_{X_j}(k) = \frac{c_j(k) + 1}{m + n}, \tag{10}$$

where $c_j(k)$ is the frequency count of k in the j th neighbor block and $m = \sum_{k \in \mathcal{A}} c_j(k)$.

To justify this choice, we consider the situation in which the encoder uses the true frequency count table of the neighbor blocks in the mixture model when encoding the current block. So, what should be done if a pixel k_0 has never been seen before. In order to encode such a pixel, it must be given a non-zero estimated probability. To overcome this, the j th PDF is calculated according to (10) with respect to the generalized Laplace law of succession. Therefore, when a pixel k_0 has never been seen before, the probability that the next pixel will be equal to k_0 is $\frac{1}{m + n}$.

In this FMM, the parameters vector $\psi = (\pi_1, \dots, \pi_K)$ contains only the mixing proportions which are calculated by maximizing the log-likelihood of the observed data X , given by:

$$l(X, \psi) = \sum_{i=1}^m \log \left(\sum_{j=1}^K \pi_j f_{X_j}(x_i) \right). \tag{11}$$

The EM-algorithm [6] seeks to find the MLE for ψ by iteratively applying the two following steps:

E-Step: with initial guesses for the parameters of the proposed mixture model $\{\pi_j^{(0)}\}_{j=1}^K$, we calculate the posterior probability $\tau_{j,i}^{(t)}$ for the observation x_i in each constituent distribution f_{X_j} at the t th iteration by:

$$\tau_{j,i}^{(t)} = \frac{\pi_j^{(t)} f_{X_j}(x_i)}{\sum_{l=1}^K \pi_l^{(t)} f_{X_l}(x_i)}, \tag{12}$$

where $i = 1, \dots, m$ and $j = 1, \dots, K$.

M-Step: we calculate $\pi_j^{(t+1)}$ at the iteration $(t + 1)$:

$$\pi_j^{(t+1)} = \frac{1}{m} \sum_{i=1}^m \tau_{j,i}^{(t)}. \tag{13}$$

With the new estimates for $\psi = (\pi_1, \dots, \pi_K)$, the E-step is repeated to recompute new posterior probabilities. The entire procedure is repeated until the model parameters converge [6, 9] to $\hat{\psi} = (\hat{\pi}_1, \dots, \hat{\pi}_K)$. Finally, the distribution used to encode the current block is a mixture calculated from its adjacent blocks distributions defined by:

$$\hat{f}_X(k) = \sum_{j=1}^K \hat{\pi}_j f_{X_j}(k), k \in \mathcal{A}. \tag{14}$$

This method needs to send $\hat{\psi}$ values as additional information to the decoder. Note that, the EM-algorithm consumes less CPU time to converge in the case of non-parametric FMM by comparison with the parametric ones. The block diagram of the proposed method is depicted in Fig. 2 where both the modeling and coding part have been included.

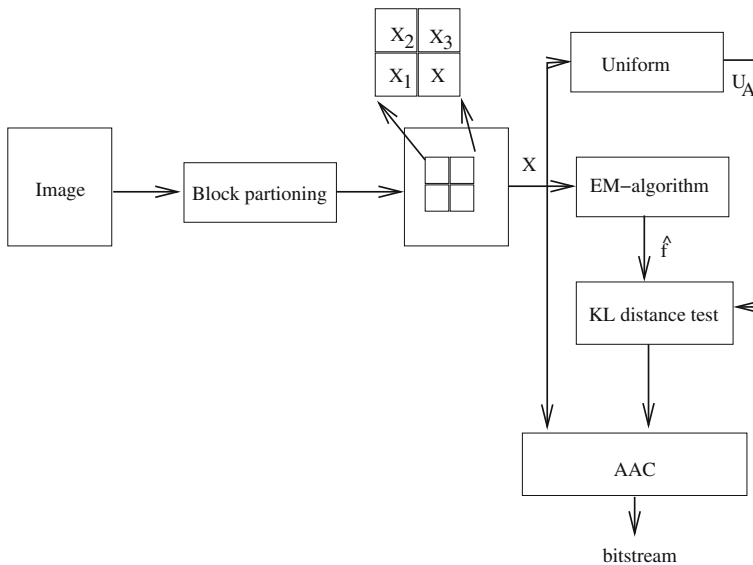


Fig. 2 Functional block diagram of the proposed approach

In practice, we use an integer implementation of AC, which works with frequency counts instead of probabilities. The estimated PDF can be written as $\hat{f}_X(k) = \frac{\hat{c}(k)}{M}$, where $\hat{c}(k) = \left\lfloor \sum_{j=1}^K \hat{\pi}_j c_j(k) \right\rfloor + 1$ is the estimated frequency count table and $M = \sum_{k \in \mathcal{A}} \hat{c}(k)$. Therefore, we propose to start encoding the current block X with the estimated frequency table \hat{c} calculated from the new mixture model. As the coding process continues, AAC updates the symbol frequency by incrementing its count just after it has been encoded, which makes the model gradually converge to the symbols' real PDF.

In this case, the probability of the current block X , which corresponds to the likelihood function given that block, is calculated as:

$$\begin{aligned} L(X, \hat{\psi}) &= \hat{f}_X(x_1) \prod_{i=2}^m \hat{f}_X(x_i | x_1, \dots, x_{i-1}) \\ &= \hat{f}_X(0)^{c(0)} \prod_{k \in \mathcal{A} \setminus \{0\}} \{\hat{f}_X(k|0, \dots, k-1)\}^{c(k)} \\ &= \frac{\prod_{k \in \mathcal{A}} \left\{ \prod_{j=1}^{c(k)} (\hat{c}(k) + j - 1) \right\}}{\prod_{i=1}^m (M + i - 1)}, \end{aligned} \quad (15)$$

and the number of bits required to represent the block X is close to $-\log_2(L(X, \hat{\psi}))$.

5 Experimental results

The AC implementation used during the experiment analysis was based on [25]. In this experimental study, we assumed that the distribution of block X , i.e the block to be encoded, is estimated by a mixture of their three causal neighboring block PDF. Therefore, from (9), the estimated distribution \hat{f}_X is:

$$\hat{f}_X = \hat{\pi}_1 f_{X_1} + \hat{\pi}_2 f_{X_2} + \hat{\pi}_3 f_{X_3}, \quad (16)$$

where f_{X_1} , f_{X_2} and f_{X_3} correspond respectively to the PDF of the blocks X_1 , X_2 and X_3 . The EM-algorithm is used to find the mixing proportions $\{\hat{\pi}_j\}_{j=1}^3$ that maximize the AC efficiency.

For our experiments, we propose to analyze our method on some well-known 8-bit grayscale images, which are available from the University of Waterloo Greyscale2 collection.¹ In Section 5.1 we analyzed and estimated the distribution of blocks from the image of Barbara and in Section 5.2, we presented and compared the compression results.

5.1 Distribution estimation

Figure 3 shows four neighbor blocks from the image of Barbara and their corresponding distributions. From Fig. 3.h we can show that the distributions f_X and \hat{f}_X are very similar. For this example, the estimated distribution is $\hat{f}_X = 0.3 f_{X_1} + 0.1 f_{X_2} + 0.6 f_{X_3}$, and the results

¹<http://links.uwaterloo.ca/repository.html>

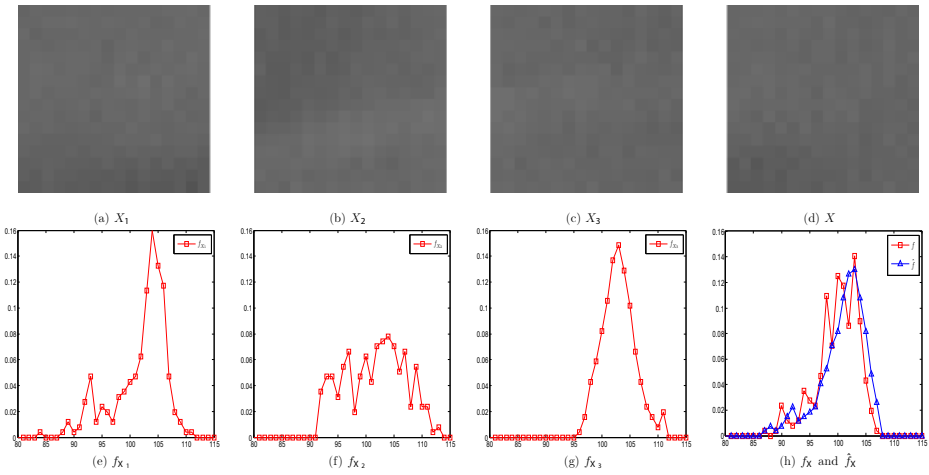


Fig. 3 Four neighbor 16×16 blocks of the image of Barbara: **a** X_1 , **b** X_2 , **c** X_3 , **d** X , block to be encoded. Their corresponding probability distributions: **e** f_{X_1} , **f** f_{X_2} , **g** f_{X_3} , **h** f_X and \hat{f}_X

of KL distances between f_X and other distributions f_{X_1} , f_{X_2} , f_{X_3} and \hat{f}_X are presented in Table 2. From these results, we found that the minimum KL distance is between f_X and \hat{f}_X , and therefore AC works better when using \hat{f}_X to encode the current block instead of using one of its neighboring blocks distributions [13].

Let U_A be a discrete uniform distribution in the alphabet set \mathcal{A} composed of n symbols and every n symbol has an equal probability $\frac{1}{n}$. If we consider a block-based AAC-0 (BAAC-0), which encodes each image block separately by starting with the discrete uniform distribution U_A and by updating the pixel frequency count just after it has been encoded, then the likelihood function is given by:

$$\begin{aligned}
 L(X, U_A) &= \frac{\prod_{k \in \mathcal{A}} \{\prod_{j=1}^{c(k)}\}}{\prod_{i=1}^m (n + i - 1)} \\
 &= \frac{\prod_{k \in \mathcal{A}} \{\prod_{j=1}^{c(k)}\}}{\prod_{i=1}^m (n + i - 1)}.
 \end{aligned}
 \tag{17}$$

Therefore, the number of bits needed to encode the block X by using BAAC-0 is bounded by $-\log_2(L(X, U_A))$.

Table 2 KL distance between f_X and some other distributions of blocks presented in Fig. 3

$KL(f_X f_{X_1})$	$KL(f_X f_{X_2})$	$KL(f_X f_{X_3})$	$KL(f_X \hat{f}_X)$
0.87	0.96	0.81	0.09

Another four neighbor blocks obtained from a Barbara image and their corresponding distributions are presented in Fig. 4. For this second example, the estimated distribution is $\hat{f}_X = 0.4f_{X_1} + 0.4f_{X_2} + 0.2f_{X_3}$. The results of KL distances between f_X and other distributions including the discrete uniform distribution are presented in Table 3. From these results, the minimum KL distance is between f_X and U_A , and therefore AC works better when using U_A to encode the current block instead of using \hat{f}_X . From Fig. 2, we can see that the proposed compression scheme adopts a modeling switcher for each block between the estimated PDF obtained by using the proposed FMM and the uniform distribution. The switcher selects modeling in uniform or in mixture mode, as determined by a simple KL distance test. Note that uniform distribution presents a good alternative for very complex models.

5.2 Compression results

In our scheme, we propose to truncate the values of the mixing proportions $\{\hat{\pi}_j\}_{j=1}^K$ (of type float) by considering one decimal point. Thus, when we have three mixing proportions for each block, we have to store only two digits because $\sum_{j=1}^3 \pi_j = 1$, which requires 8 bits.

Note that, we have experimentally found that truncating the mixing proportions up to one decimal point does not give any improvement in the overall compression rate. Therefore, by using three mixing proportions and the PDF of causal neighboring blocks, we can estimate the PDF of each block. In addition, it should be noticed that the proposed compression algorithm can be applied with the same manner in both pixel and prediction error domains. Without loss of generality, we employ in this paper the median edge detector (MED), which is a nonlinear predictor used in LOCO-I image coder [24], to calculate the predicted errors according to following equations:

$$\hat{x} = \begin{cases} \min(a, b) & \text{if } C \geq \max(a, b) \\ \max(a, b) & \text{if } C \leq \min(a, b) \\ a + b - c & \text{otherwise} \end{cases} \quad (18)$$

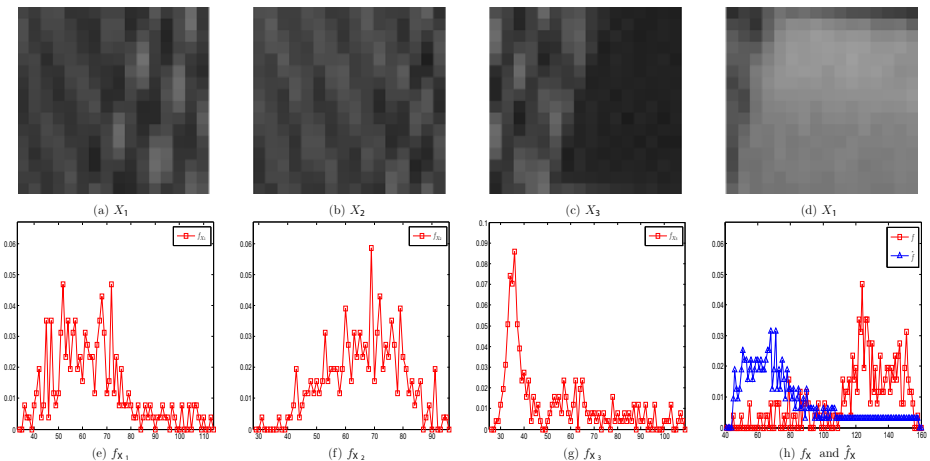


Fig. 4 Four neighbor 16×16 blocks of the image of Barbara: **a** X_1 , **b** X_2 , **c** X_3 , **d** X , block to be encoded. Their corresponding probability distributions: **e** f_{X_1} , **f** f_{X_2} , **g** f_{X_3} , **h** f_X and \hat{f}_X

Table 3 KL distance between f_X and some other distributions of blocks presented in Fig. 4

$KL(f_X f_{X_1})$	$KL(f_X f_{X_2})$	$KL(f_X f_{X_3})$	$KL(f_X \hat{f}_X)$	$KL(f_X U_{\mathcal{A}})$
1.71	1.71	1.78	1.18	0.5

The MED predictor can be viewed as the median selector of three predictors; a , b and c ; where the set $\{a, b, c\}$ is a 3-pixel neighborhood for the prediction of the current pixel. The predicted error pixel is calculated as $e = x - \hat{x}$ and will be mapped to a non-negative value by applying successively (19) and (20) in order to improve the AC efficiency by reducing the error range from $[-255, 255]$ to $[0, 255]$. The mapping process leads to a model which better fit the source symbols statistics.

$$\begin{cases} e = e + 256 & \text{if } e < -128 \\ e = e - 256 & \text{if } e > 128 \end{cases} \quad (19)$$

$$\begin{cases} e = 2|e| & \text{if } e \geq 0 \\ e = 2|e| - 1 & \text{if } e < 0 \end{cases} \quad (20)$$

Table 4 lists the compression results, in bits per pixel (bpp), measured by the state-of-the-art lossless image compression standards and algorithms. The columns correspond respectively to:

- **FELICS** [7]: The compression mechanism of Howard and Vitter.
- **JPEG-2000** [22]: An international standard for still image compression, which supports a wider set of features.
- **bzip2** [4]: A highly-quality implementation of the Burrows-Wheeler transform.
- **JPEG-LS** [24]: The latest coding standard for lossless and near-lossless of continuous-tone images.
- **BBAAC method** [13]: A new AAC which estimates the probability distribution of each block of pixels by minimizing the Kullback-Leibler distance between the exact PDF of that block and the PDF of its causal neighbor blocks.
- **CALIC** [26]: The context-based adaptive lossless image codec, was evaluated to be in the best performance by the JPEG working group.
- **Proposed approach in pixel domain (PD)**: Proposed approach applied in pixel domain with blocks of size 16×16 .
- **Proposed approach in prediction error domain (PED)**: Proposed approach applied in prediction error domain by applying the MED predictor and with blocks of size 16×16 .

CALIC, JPEG-2000 and JPEG-LS are among well known lossless image compression algorithms. Among these CALIC was evaluated to be in best performance by the JPEG working group.

The average compression rate obtained by encoding each block, of size 16×16 , with a mixture of three causal block PDF in PD is 4.94 bpp , while the average coding rate in PED when using the MED predictor is 4.51 bpp , which outperforms JPEG-LS by 2.6% . In addition, from the Table 4, it is apparent that the proposed compression scheme, if we mix PD and PED, offers fairly competitive performance to CALIC, with only 2% increase in average bit-rate.

Table 5 presents the best coding rates attained by a hybrid system that applies both alternatives, AC of sources in PD and PED. These coding rates are obtained from two different

Table 4 Compression results of the state-of-the-art lossless image compression standards and algorithms on Waterloo image test set in bits/pixel (bpp)

Name	FELICS	BZIP2	BBAAC	JPEG2000	JPEG-LS	CALIC	Proposed approach	
							PD	PED
Barbara	5.34	6.12	6.46	4.66	4.73	4.45	5.96	4.96
Boat	4.67	5.33	5.79	4.40	4.25	4.15	5.36	4.53
France	2.36	0.39	3.20	2.02	1.41	0.82	3.05	0.67
Frog	6.13	3.46	3.73	6.25	6.05	5.85	3.57	5.29
Goldhill	4.97	5.60	6.26	4.83	4.71	4.63	5.85	4.85
Lena	4.60	5.30	5.99	4.31	4.24	4.11	5.47	4.45
Library	5.47	4.48	4.50	5.69	5.10	5.01	4.45	5.24
Mandrill	6.30	6.60	6.90	6.11	6.04	5.87	6.64	6.19
Mountain	6.57	5.13	5.08	6.70	6.42	6.27	5.08	6.53
Peppers	4.79	5.34	6.10	4.61	4.49	4.38	5.61	4.84
Washsat	4.15	2.52	3.01	4.43	4.13	3.67	2.71	2.31
Zelda	4.29	5.12	6.19	3.99	4.01	3.86	5.57	4.22
Average	4.97	4.62	5.27	4.83	4.63	4.42	4.94	4.51

The notations PD and PED means Pixel Domain and Prediction Error Domain, respectively

sets of results, beating JPEG-LS and CALIC by 9.7 and 5.4 %, respectively. We plan to continue exploring this approach to provide a single system for an appropriate choice of source basis before doing AC process.

Table 5 A mix system which chooses for each image the best compression result between PD and PED

Name	JPEG-LS	CALIC	Proposed approach		Mixing
			PD	PED	
Barbara	4.73	4.45	5.96	4.96	4.96
Boat	4.25	4.15	5.36	4.53	4.53
France	1.41	0.82	3.05	0.67	0.67
Frog	6.05	5.85	3.57	5.29	3.57
Goldhill	4.71	4.63	5.85	4.85	4.85
Lena	4.24	4.11	5.47	4.45	4.45
Library	5.10	5.01	4.45	5.24	4.45
Mandrill	6.04	5.87	6.64	6.19	6.19
Mountain	6.42	6.27	5.08	6.53	5.08
Peppers	4.49	4.38	5.61	4.84	4.84
Washsat	4.13	3.67	2.71	2.31	2.31
Zelda	4.01	3.86	5.57	4.22	4.22
Average	4.63	4.42	4.94	4.51	4.18

6 Conclusions

In this work, we have proposed an efficient scheme for block-based lossless image compression using adaptive arithmetic coding. The model proposed to estimate the statistics of each block is designed by a finite mixture model of non-parametric distributions by exploiting the high correlation between neighboring blocks. The experimental results showed that the compression efficiency of arithmetic coding when using finite mixture model for data statistics estimation is increased, and when combined with the median edge detector, it outperforms JPEG-LS by 2.6 %. An hybrid algorithm combining the results obtained when encoding image in PD and PED does very well, beating JPEG-LS by 9.7 %. More predictors should be studied and a single system which switch between pixel and prediction error is needed to provide more improvement.

References

1. Abramson N (1963) Information theory and coding. McGraw-Hill Book Company, Inc., New York
2. Alcaraz-Corona S, Rodriguez-Dagnino RM (2010) Bi-level image compression estimating the markov order of dependencies. *IEEE J Sel Topics Signal Process* 4(3):605–611
3. Auli-Llinas F (2011) Stationary probability model for bitplane image coding through local average of wavelet coefficients. *IEEE Trans Image Process* 20(8):2153–2165
4. Burrows M, Wheeler DJ (1994) A block-sorting lossless data compression algorithm, Tech. Rep., digital equipment corporation, Palo Alto, California
5. Carpentieri B (1997) A new lossless image compression algorithm based on arithmetic coding. In: Proceedings of the 9th international conference on image analysis and processing-volume II, pp 54–61
6. Dempster AP, Laird NM, Rubin DB (1977) Maximum likelihood from incomplete data via the em algorithm. *J R Stat Soc Ser B* 39(1):1–38
7. Howard PG, Vitter JS (1993) Fast and efficient lossless image compression. In: Proceedings 1993 data compression conference. Mars, IEEE Computer Society Press, pp 351–360
8. Howard PG, Vitter JS (1994) Arithmetic Coding for Data Compression. *Proc IEEE* 82(6):857–865
9. Jeff Wu CF (1983) On the convergence properties of the EM algorithm. *Ann Stat* 11(1):95–103
10. Kullback S, Leibler RA (1951) On information and sufficiency. *Ann Math Stat* 22:49–86
11. Kuroki N, Manabe T, Numa M (2004) Adaptive arithmetic coding for image prediction errors. In: Proceedings of IEEE international symposium on circuits and systems (ISCAS 04), vol III, pp 961–964
12. Langdon GG (1984) An Introduction to Arithmetic Coding. *IBM J Res Dev* 28:2
13. Masmoudi A, Masmoudi A (2013) A new arithmetic coding model for a block-based lossless image compression based on exploiting inter-block correlation. *Sig Image Video Process* 1–7
14. Masmoudi A, Puech W, Bouhlef MS (2010) Efficient adaptive arithmetic coding based on updated probability distribution for lossless image compression. *J Electron Imaging* 19(2):023014
15. Matsuda I, Shirai N, Itoh S (2003) Lossless coding using predictors and arithmetic code optimized for each image. In: Proceedings of international workshop visual content processing and representation, vol 2849, pp 199–207
16. McLachlan GP, Peel D (2000) Finite mixture models, wiley series in probability and statistics. Wiley-Interscience, 1st edn
17. Minoo K, Nguyen TQ (2009) Entropy coding via parametric source model with applications in fast and efficient compression of image and video data. In: DCC'09
18. Moffat A, Neal RM, Witten IH (1998) Arithmetic coding revisited. *ACM Trans Inf Syst* 16(3):256–294
19. Rubin F (1979) Arithmetic stream coding using fixed precision registers. In: Data compression conference, pp 672–675. IEEE transaction on information theory
20. Salomon D (2007) Data compression the complete reference fourth edition. Springer
21. Shannon CE (1948) A mathematical theory of communication. *Bell Syst Tech J* 27:379–423
22. Taubman DS, Marcellin MW (2002) JPEG2000: image compression fundamentals; standards and practice. Springer
23. Teuhola J (2011) Nonblock coding of binary sources by probabilistic enumeration. *IEEE Trans Inf Theory* 57(9):6170–6179

24. Weinberger MJ, Seroussi G, Sapiro G (2000) The LOCO-I lossless image compression algorithm: principles and standardization into JPEG-LS. *IEEE Trans Image Process* 9(8):1309–1324
25. Witten IH, Neal RM, Cleary JG (1987) Arithmetic coding for data compression. *Commun ACM* 30(6):520–540
26. Wu X, Memon N (1996) CALIC - context based adaptive lossless image codec. *IEEE Int Conf Acoust Speech Sig Process* 4:1890–1893
27. Wu J, Xu Z, Jeon G, Zhang X, Jiao L (2013) Arithmetic coding for image compression with adaptive weight-context classification. *Sig Process Image Commun* 28(7):727–735
28. Ye H, Deng G, Devlin JC (2002) Parametric probability models for lossless coding of natural images. In *EUSIPCO*, pp 514–517
29. Zhang L, Wang D, Zheng D (2012) Segmentation of source symbols for adaptive arithmetic coding. *IEEE Trans Broadcast* 58(2):228–235
30. Zhao DY, Samuelsson J, Nilsson M (2008) On entropy-constrained vector quantization using gaussian mixture models. *IEEE Trans Commun* 56(12):2094–2104



Atef Masmoudi received his engineering diploma in informatics from the National School of Computer Science in 2003 and a master's in automatic and signal processing from the National Engineering School of Tunisia in 2005. He received his PhD degree in computer science from the University of Montpellier II, France, in 2010.

From 2005 to 2010, he was an assistant in computer science at the University of Sfax, Tunisia, and since 2010 he was been an assistant professor. From 2003 to 2012, he was a member of the Research Unit: Sciences and Technologies of Image and Telecommunications, Tunisia, and from 2007 to 2010 he was a member of the LIRMM Laboratory (Laboratory of Informatics, Robotics, and Microelectronics of Montpellier), France. He was a member of the organization committee of the IEEE International Conference: Sciences of Electronic, Technologies of Information and Telecommunication in 2005, 2007, and 2009. His main research area is the combination of compression and encryption by using arithmetic coding and chaos theory. He has published more than 6 journal papers, and more than 10 conference papers.



William Puech received the diploma of Electrical Engineering from the University of Montpellier, France, in 1991 and the Ph.D. Degree in Signal-Image-Speech from the Polytechnic National Institute of Grenoble, France in 1997. He started his research activities in image processing and computer vision. He served as a Visiting Research Associate to the University of Thessaloniki, Greece. From 1997 to 2000, he had been an Assistant Professor in the University of Toulon, France, with research interests including methods of active contours applied to medical images sequences. Between 2000 and 2008, he had been Associate Professor and since 2009, he is full Professor in image processing at the University of Montpellier, France. He works in the LIRMM Laboratory (Laboratory of Computer Science, Robotic and Microelectronic of Montpellier). His current interests are in the areas of protection of visual data (image, video and 3D object) for safe transfer by combining watermarking, data hiding, compression and cryptography. He has applications on medical images, cultural heritage and video surveillance. He is the head of the ICAR team (Image & Interaction) and he has published more than 15 journal papers, 8 book chapters and more than 80 conference papers. W. Puech is associate editor of J. of Advances in Signal Processing, Springer and he is reviewer for more than 15 journals (IEEE Trans. on Image Processing, IEEE Trans. on Multimedia, IEEE TCSVT, IEEE TIFS, Signal Processing: Image Communication, ...) and for more than 10 conferences (IEEE ICIP, EUSIPCO, ...).



Afif Masmoudi is a Professor in Mathematics. He is a member of the Research Unit: Probability and Statistics. He received his HDR in Statistics from the University of Sfax, Tunisia, in 2005.

His main research area is the parametric and nonparametric estimation, implicit estimation, asymptotic properties of variance function, learning Bayesian network. He has published more than 18 journal papers, and more than 15 conference papers.