J Digit Imaging (2016) 29:706–715
DOI 10.1007/s10278-016-9892-y

Novel Near-Lossless Compression Algorithm for Medical Sequence Images with Adaptive Block-Based Spatial Prediction

Xiaoying Song[1] · Qijun Huang[1] · Sheng Chang[1] · Jin He[1] · Hao Wang[1]

Published online: 14 July 2016

Abstract To address the low compression efficiency of lossless compression and the low image quality of general near-lossless compression, a novel near-lossless compression algorithm based on adaptive spatial prediction is proposed for medical sequence images for possible diagnostic use in this paper. The proposed method employs adaptive block size-based spatial prediction to predict blocks directly in the spatial domain and Lossless Hadamard Transform before quantization to improve the quality of reconstructed images. The block-based prediction breaks the pixel neighborhood constraint and takes full advantage of the local spatial correlations found in medical images. The adaptive block size guarantees a more rational division of images and the improved use of the local structure. The results indicate that the proposed algorithm can efficiently compress medical images and produces a better peak signal-to-noise ratio (PSNR) under the same pre-defined distortion than other near-lossless methods.

✉ Qijun Huang
huangqj@whu.edu.cn

Xiaoying Song
xiaoying811@whu.edu.cn

Sheng Chang
changsheng@whu.edu.cn

Jin He
jin.he@whu.edu.cn

Hao Wang
wanghao@whu.edu.cn

[1] School of Physics and Technology, Wuhan University, Wuhan, Hubei, China 430072

Keywords Adaptive block size · Spatial prediction · Block searching · Lossless Hadamard transform · Near-lossless compression

Introduction

Large amounts of image data are produced in the field of medical imaging, especially computed tomography (CT) and magnetic resonance images (MRI), which are always a succession of 2D images (slices). The amount of data generated may be so large that it results in impractical storage, processing, and communication requirements [1]. Image compression solves these problems. Lossless compression methods are traditionally used to avoid losing any critical diagnostic information. Although lossless methods are preferable for medical images, the compression ratios achieved using these methods are rather modest, ranging from 2 to 4 depending on the images and the methods employed. Thus, many researchers in the medical image compression community argue that some distortion must be allowed in the reconstructed images to obtain higher compression ratios. These researchers argue that the use of lossy compression in diagnostic imaging is justified only if diagnostically important information is preserved [2]. To ensure the high diagnostic quality of compressed images, very little distortion can be allowed. Methods with these characteristics are defined as "near-lossless" compression. The definition of near-lossless compression, first established in the field of medical imaging [3], assumes that the peak of absolute error (PAE) between the original and the reconstructed image should be user-defined.

Many block-based compression algorithms used on still images are based on transforms, that is, they are performed

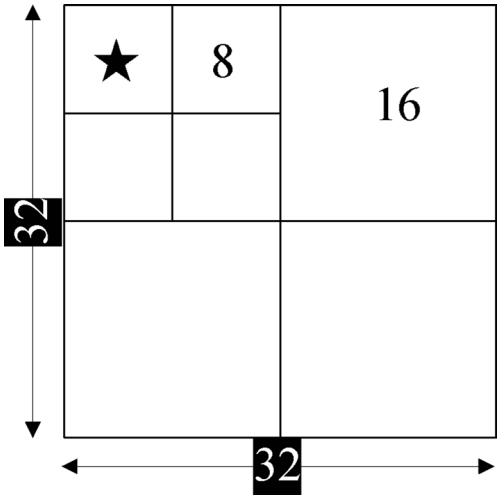

Fig. 1 Division of the first 32 × 32 block

in the frequency domain, such as discrete cosine transform (DCT) and discrete wavelet transform (DWT). Singh et al.

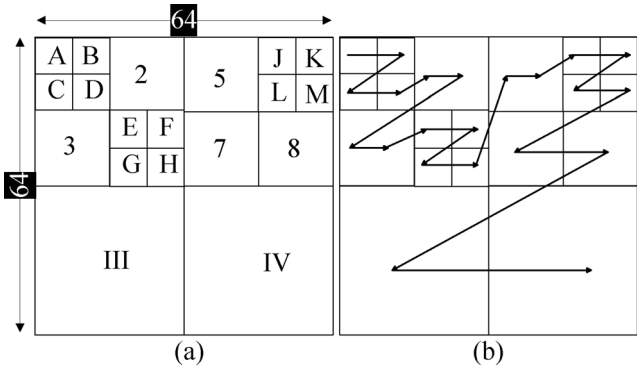

Fig. 2 **a** A demonstration of block division. **b** The corresponding scanning order of **a**

[4] proposed an adaptive threshold-based block classification for DCT in medical image compression. Lee et al. [5] proposed a frame rate conversion method using adaptive block sizes and search windows corresponding to motion activity levels. Muhit et al. [6] proposed a block-partitioning scheme that incorporates both geometry-adaptive partitioning and an elastic motion model as extensions to the standard procedure for motion estimation. Zhao et al. [7] developed a super-spatial prediction method for structural regions of natural images. Medical images have two characteristics that differ from natural images: they contain many structural similarities and symmetric characteristics resulting from inherent symmetry of human anatomy, and there is usually a large amount of diagnostically useless dark background in medical images, with pixel values approximately 0. Thus, both in the object region and in the background, medical images contain many local spatial correlations. Most block-based methods are constructed in the frequency domain, which not only fails to utilize local spatial redundancy but also has high computational complexity.

One of the most popular methods used in near-lossless systems is predictive coding, such as differential pulse code modulation (DPCM) [8]. Aràndiga et al. [9] proposed a multiscale compression algorithm within Harten's interpolatory framework for multiresolution, which yields a specific estimate of precise error between the original and the decoded signal. Caldelli et al. [10] used JPEG-LS [11] to design a near-lossless system, which is expressly used in remote sensing and telemedicine applications. Miguel et al. [12] compressed prediction residuals with a near-lossless bit plane coder for hyperspectral images. Wavelet-based methods have also been used in near-lossless compression [13]. In these near-lossless methods, the peak signal-to-noise ratio (PSNR) decreases rapidly as PAE increases. The high quality of medical images must be preserved, so in this paper, we propose a near-lossless method that can guarantee a high PSNR even when PAE is large.

Referring to the above idea of block-based prediction and the idea of motion estimation from video coding, we propose an adaptive block size-based spatial prediction in 2-D spaces. Through block searching and matching, an image is predicted directly in the spatial domain in a block-by-block basis. The block size is adaptively selected according to a Lagrange cost function. Block-based spatial prediction can effectively explore local spatial relationships, which is appropriate for the characteristics of medical images. An adaptive block size guarantees a more rational division of images and the improved use of local

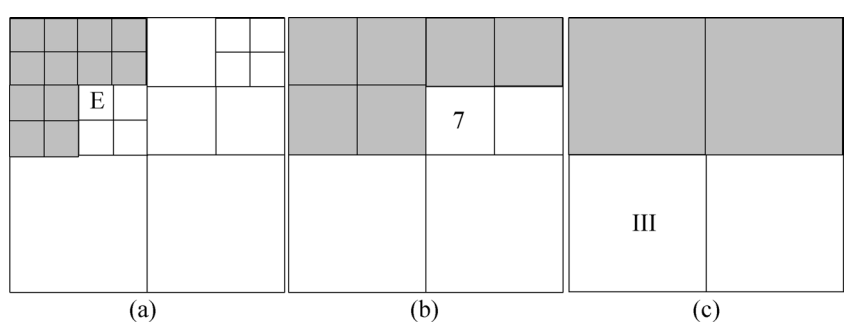

Fig. 3 Adaptive reference blocks with different sizes. **a** The size of block "E" is 8 × 8. **b** The size of block "7" is 16 × 16. **c** The size of block "III" is 32 × 32

structure for different types of images. During the block searching process, an improved search method is proposed instead of using the full search to improve searching speed, which will be described in more detail later in the paper. After block searching, prediction residuals are transformed by Lossless Hadamard Transform before quantization to obtain a high quality reconstructed image. Extensive experimental results demonstrate that the proposed method achieves good compression performance and outperforms some other near-lossless methods in terms of PSNR.

The remainder of the paper is organized as follows. The proposed block-based algorithm is described in Section 2. Near-lossless results are compared with the results of other algorithms in Section 3. Finally, Section 4 presents the paper's conclusions.

Proposed Method

This section presents the proposed compression algorithm in detail. In the proposed method, initially, an input image is segmented into non-overlapping blocks with a maximum block size of 32 × 32. Each block is predicted using any previously encoded blocks; meanwhile, some transformation templates are defined during block matching to obtain better prediction results. According to the Lagrange cost function, if treating a block as a whole does not result in good performance, it will be quadtree partitioned into four small subblocks. This process proceeds iteratively until the minimum block size is reached. Then, the prediction residuals are transformed using a Lossless Hadamard Transform to further remove statistical redundancy, and quantization is performed on the transformed coefficients to achieve near-lossless compression. Finally, the quantized coefficients are entropy coded. Details of each stage of the method will be discussed in the following subsections.

Table 1 Eight transformation templates

No.	Geometric operation
T1	Vertical flip
T2	Horizontal flip
T3	Left rotation (90°)
T4	Right rotation (90°)
T5	Rotation (180°)
T6	Diagonal flip along 135°
T7	Diagonal flip along 45°
T8	No operation

Adaptive Block Size-Based Prediction

The primary question of block-based prediction is how large the block size should be. When the block size is large, complex textures or distortions in structural regions or regions with a variety of content will not be captured well. When the block size is small, performance can improve, but complexity is higher and the searching process can be longer. To optimize these conditions, we use an adaptive block size. The maximum block size is set as 32 × 32 instead of 16 × 16, as specified in H.264, because medical images have wider flat regions than video images. Research has shown that the minimum block size of 8 × 8 is a reasonable choice in terms of coding performance, and we therefore adopt this minimum block size.

The first 32 × 32 block, which has no reference blocks, will be directly stored as prediction residuals. Considering its large amount of data, it is further hierarchically divided

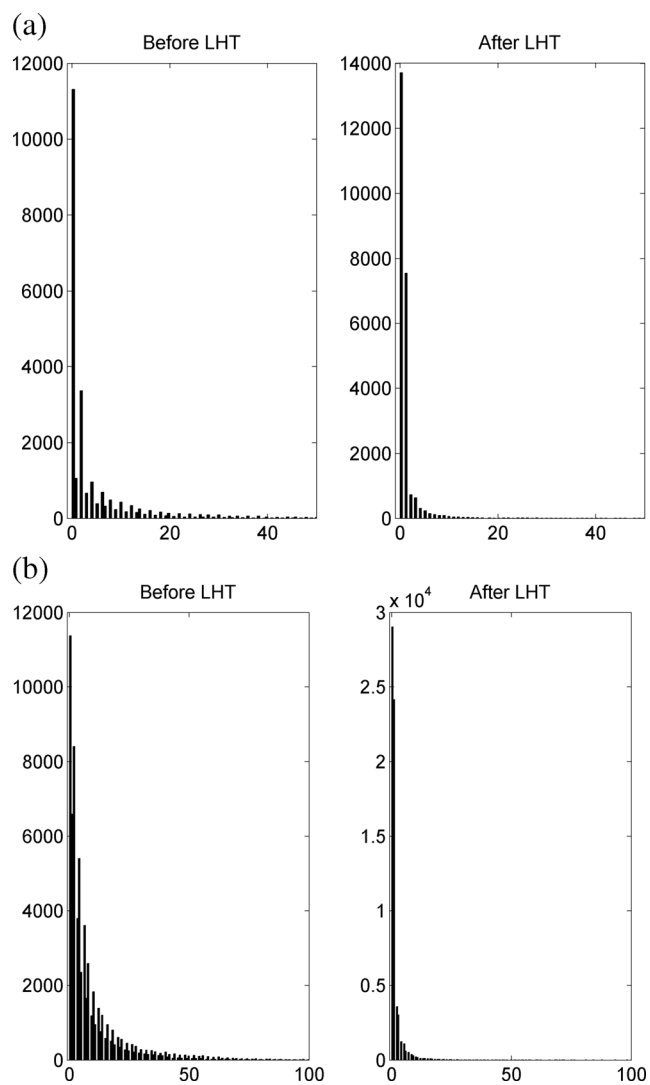

Fig. 4 Histograms before LHT and after LHT. **a** CT_Aperts **b** MRI_liver_t1

Table 2 Descriptions of the medical sequence images used

Databases	File name	History	Age	Sex	Voxel size (mm)	Volume size
CIPR	CT_Aperts	Apert's syndrome	2	M	035 × 0.35 × 2	256 × 256 × 97
	CT_carotid	Internal carotid dissection	41	F	025 × 0.25 × 1	256 × 256 × 74
	CT_skull	Tripod fracture	16	M	070 × 0.70 × 2	256 × 256 × 203
	CT_wrist	Healing scaphoid fracture	20	M	070 × 0.70 × 2	256 × 256 × 183
	MR_liver_t1	Normal	38	F	145 × 1.45 × 5	256 × 256 × 58
	MR_liver_t2e1	Normal	38	F	137 × 1.37 × 5	256 × 256 × 58
	MR_ped_chest	Congenital heart disease	1	M	078 × 0.78 × 5	256 × 256 × 77
	MR_sag_head	Left exophthalmos	42	M	098 × 0.98 × 3	256 × 256 × 58
CVG	Mr_brain	–	–	–	–	256 × 256 × 16

as shown in Fig. 1. The final first block is indicated by a five-pointed star, whose size is reduced to the minimum block size.

Any 32 × 32 block may be predicted as a whole block or may be quadtree partitioned into four subblocks 16 × 16 in size. Any 16 × 16 block may be predicted as a whole block or may be quadtree partitioned into four subblocks 8 × 8 in size. The choice to divide a block is performed by selecting the block size that minimizes the Lagrange cost function, which is given by:

$$J = D + R \tag{1}$$

Here, D is the sum of the absolute residuals between the original and the reconstructed pixels within the current block, and R is the number of bits required to transmit the block. If the sum, J, of the four quadtree partitioned subblocks is smaller, division is performed.

Because the block size is adaptive, the sizes of corresponding reference blocks should also be adaptive, and the searching process should obey certain orders. For any current block, all of the encoded blocks are partitioned or merged into blocks of the same size as its reference blocks. Blocks of size 32 × 32 are processed in raster-scanning order; inside a block, its subblocks of the same level are also scanned in raster-scanning order. Figure 2a shows an example of block division and Fig. 2b indicates its corresponding scanning order. Figure 3 shows some examples of adaptive reference blocks with different sizes, which are indicated in gray.

Block Searching Process

The smaller the values of the prediction residuals, the better the compression performance that can be obtained. The values of the prediction residuals depend on the effectiveness of block matching. To describe the spatial relationships between blocks in detail, eight transformation templates are defined and presented in Table 1. These templates depict all available block transformations and efficiently describe their spatial and structural similarities.

Blocks are predicted using any previously encoded blocks after undergoing the above-mentioned transformations. For a current block b_c, the block resulting in the

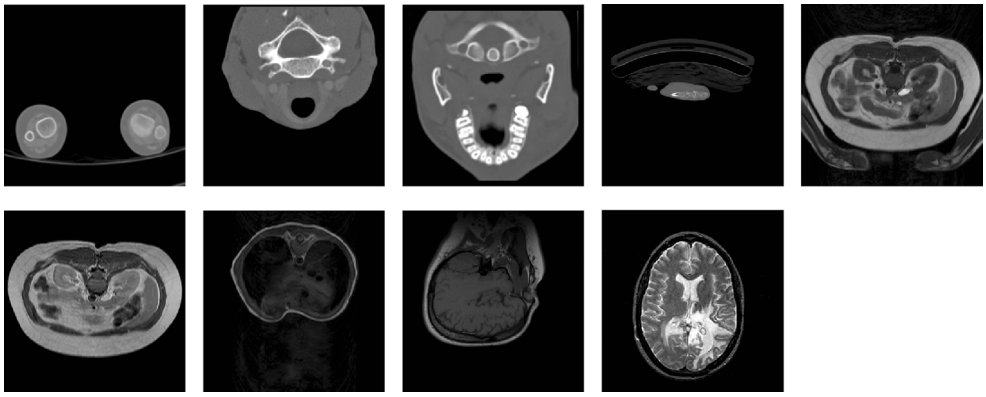

Fig. 5 Descriptions of medical sequence images: first slice of each data set. **a** CT_Aperts. **b** CT_carotid. **c** CT_skull. **d** CT_wrist. **e** MR_liver_t1. **f** MR_liver_t2e1. **g** MR_ped_chest. **h** MR_sag_head. **i** Mr_brain

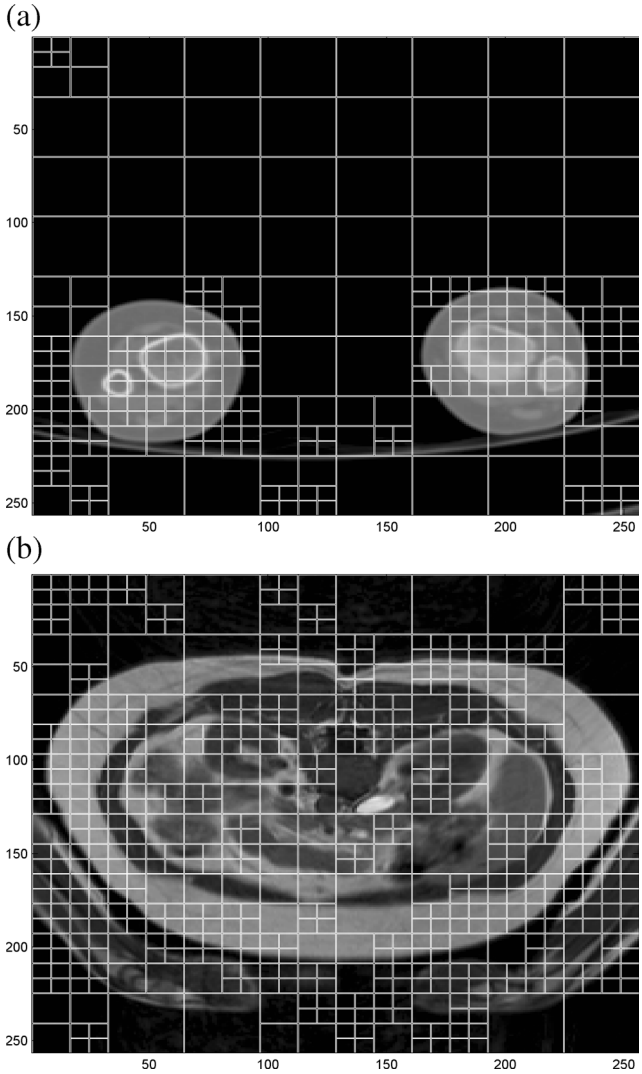

Fig. 6 Adaptive partitioning maps. **a** CT_Aperts **b** MRI_liver_t1

minimum matching difference is selected as the optimal reference block. The sum of the absolute difference (SAD) is used to measure the matching difference. We assume that the size of block b_c is M × M and that b_m denotes one of the reference blocks. With $m = (i,j)$ as the blocks' relative displacement, the SAD between block b_c and b_m is calculated as follows:

$$SAD(m) = \sum_{x=1}^{M} \sum_{y=1}^{M} |b_c(x,y) - b_m(x+i, y+j)| \tag{2}$$

Because structural components may be repeated or distorted not only in neighboring domains but also at symmetrical positions, the full search should be applied for block matching. However, the full search is time-consuming. Many fast algorithms, such as three step search [14] and cross search [15], search only a small subset of available reference blocks; consequently, the optimal matching block found is a local rather than global optimum. Moshnyaga et al. [16] presented a progressive block matching method in which all reference blocks are processed in parallel and matches are iterated row by row. Despite the decline in the total number of operations, comparisons are performed more than once for each block. The characteristics of SAD were used to reduce computational complexity in ([17, 18]). If one reference block does not match the SAD conditions, it is omitted directly. In this paper, we use SAD characteristics to improve the block searching speed. Then, Eq. (2) becomes:

$$SAD(m) = |BC - BM| \tag{3}$$

$$BC = \sum_{x=1}^{M} \sum_{y=1}^{M} b_c(x,y) \tag{4}$$

$$BM = \sum_{x=1}^{M} \sum_{y=1}^{M} b_m(x+i, y+j) \tag{5}$$

BC and *BM* denote the sums of pixel values within blocks b_c and b_m, respectively. According to the property of absolute inequality, we obtain

$$BC - SAD(m) \le BM \le BC + SAD(m) \tag{6}$$

Assuming that we have obtained a $SAD(m)$ from reference block b_m. If another reference block, b_n, is a better choice, it must satisfy the condition $SAD(n) \le SAD(m)$. Thus, block b_n should be searched only if the following inequality holds:

$$BC - SAD(m) \le BN \le BC + SAD(m) \tag{7}$$

Similarly, *BN* denotes the sum of pixel values within block b_n. Because *BN* is a constant value independent of which transformation has been performed, for each reference block, we first determine whether it satisfies the inequality (7). If condition (7) holds, the current best matching block b_m would be replaced by block b_n. If not, block b_n would be skipped, the eight types of transformation would be omitted, and then the time spent on searching would be saved.

Lossless Hadamard Transform and Quantization

The prediction residuals obtained from perfect prediction should not contain any correlations. However, the residuals are still spatially correlated to a certain extent. To further sharpen the histogram distribution before entropy coding, and therefore to improve the efficiency of the proposed algorithm, the prediction residuals are transformed by Lossless Hadamard Transform (LHT). Each element of the Hadamard transformation matrix is either plus or minus if the normalization factor is ignored, which is beneficial for implementation. However, for a 4 × 4 block, the dynamic range of the coefficients will be 16 times larger than that

Table 3 Comparison of bit rates (in bpp) for adaptive and fixed block sizes

	Image sets	8 × 8	16 × 16	32 × 32	Adaptive size
CT	CT_Aperts	1.583	1.664	1.783	1.562
	CT_carotid	2.431	2.541	2.705	2.406
	CT_skull	3.368	3.491	3.646	3.356
	CT_wrist	2.398	2.465	2.602	2.364
Average		2.445	2.540	2.684	2.422
MRI	MR_liver_t1	4.139	4.225	4.378	4.090
	MR_liver_t2e1	3.081	3.127	3.266	3.004
	MR_ped_chest	3.765	3.802	3.890	3.724
	MR_sag_head	3.450	3.515	3.599	3.428
	mr030	2.998	3.096	3.211	2.938
Average		3.487	3.553	3.669	3.437
Average (CT, MRI)		3.024	3.103	3.231	2.986

Average rows are in gray.

of the input. Therefore, we decompose the coefficients into integers and remainders by dividing them by 16 and use four bits to represent the remainders. Due to the inherent redundancies in the last four bits of the transformed coefficients, the remainder can be represented by two bits [19], each of which contains all of the information of the original two bits. Take a 4 × 4 block for example.

$$\begin{bmatrix} 0 & 0 & 0 & 12 \\ 0 & 22 & 68 & 122 \\ 68 & 132 & 186 & 226 \\ 182 & 228 & 248 & 250 \end{bmatrix} \xrightarrow{2D-LHT} \begin{bmatrix} 1744 & -480 & -24 & -240 \\ -1296 & 120 & 112 & 64 \\ 96 & 280 & -40 & 120 \\ -496 & 32 & 0 & 8 \end{bmatrix} \xrightarrow{\substack{Truncated\ into\ integer \\ and\ residual}} \begin{bmatrix} 109 & -30 & -2 & -15 \\ -81 & 7 & 7 & 4 \\ 6 & 17 & -3 & 7 \\ -31 & 2 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 8 & 0 \\ 0 & 8 & 0 & 0 \\ 0 & 8 & 8 & 8 \\ 0 & 0 & 0 & 8 \end{bmatrix} \xrightarrow{\substack{remove \\ redundancy\ of \\ redisual}} \begin{bmatrix} 109 & -30 & -2 & -15 \\ -81 & 7 & 7 & 4 \\ 6 & 17 & -3 & 7 \\ -31 & 2 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 2 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 2 & 2 & 2 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (8)$$

Figure 4 shows the histogram of prediction residuals of two sample images. After LHT, the distribution of coefficients is much sharper and closer to 0, so the range of residuals is narrowed efficiently.

In this study, we quantize the two-bit remainders of the transformed coefficients to achieve near-lossless compression. A direct method of discarding bits is used to implement quantization. When the last bits of the remainders are discarded, we discard the last two bits of the original coefficients. The corresponding PAE is 3; we then use (PAE + 1)/2 to modify the remainders to decrease the PAE, and the final PAE is 2. When two bits of remainder are discarded, we discard the last four bits of the original coefficients. The corresponding PAE is 15, which after modification becomes 8. The near-lossless results of this method are compared with those of other algorithms in the next section.

Entropy Coding

Our proposed algorithm uses combined entropy coding. Spatial redundancy can be removed by prediction. Similarly, statistical redundancy can be removed by entropy coding the residuals [20]. Entropy coding allots fewer bits to higher frequency symbols and more bits to lower frequency symbols. Arithmetic coding is frequently used for entropy coding. However, the initial residuals range from −256 to 255, or 0 to 255 after remapping, which is still very large. Directly using the residuals for arithmetic coding is difficult. To avoid this implementation issue, binary arithmetic coding is adopted. The simple method of exp-Golomb coding is used for binarization. Specifically, we use the adaptive exp-Golomb coding proposed by [21]. The level of the exp-Golomb coder is selected adaptively

Table 4 Comparison of encoding time (in seconds) for improved search and full search

	Image	Full search	Improved search	Percentage time saved
CT	CT_Aperts	25.38	4.25	83 %
	CT_carotid	19.54	4.28	78 %
	CT_skull	39.59	6.22	84 %
	CT_wrist	10.04	2.81	72 %
MRI	MR_liver_t1	61.40	8.79	86 %
	MR_liver_t2e1	33.88	5.78	83 %
	MR_ped_chest	37.60	6.46	83 %
	MR_sag_head	24.13	4.55	81 %
	mr030	34.85	5.02	86 %

according to the input symbol. To enhance the efficiency of binary arithmetic coding, a generic order 3 arithmetic coder [22] is used to encode the binary residuals. Finally, the encoded bit stream is set to the decoder.

Table 5 Comparison of bit rates (in bpp) and PSNR (in dB) for the proposed method and some lossless methods

	Image sets	CALIC	JPEG-LS	JPEG2000	SPIHT	Proposed				
						Lossless	PAE=2		PAE=8	
						bpp	bpp	PSNR	bpp	PSNR
CT	CT_Aperts	1.178	0.984	1.271	2.365	1.562	1.269	61.04	1.054	48.68
	CT_carotid	1.817	1.764	2.030	3.274	2.406	1.935	60.33	1.546	48.52
	CT_skull	2.785	2.549	3.001	4.375	3.356	2.725	58.98	2.160	47.07
	CT_wrist	1.780	1.515	1.767	3.148	2.364	1.855	60.27	1.458	48.53
Average		1.890	1.703	2.017	3.291	2.422	1.946	60.16	1.555	48.20
MRI	MR_liver_t1	3.249	3.148	3.266	4.704	4.090	3.210	58.40	2.535	46.78
	MR_liver_t2e1	2.413	2.410	2.582	3.731	3.004	2.494	60.07	1.999	48.37
	MR_ped_chest	3.035	2.930	3.031	4.493	3.724	2.867	58.47	2.250	47.04
	MR_sag_head	2.598	2.573	2.915	4.336	3.428	2.814	58.73	2.314	46.62
	mr030	2.172	2.154	2.484	3.771	2.938	2.612	56.58	2.228	44.20
Average		2.693	2.643	2.856	4.207	3.437	2.799	58.45	2.265	46.60
Average(CT, MRI)		2.336	2.225	2.483	3.800	2.986	2.420	59.21	1.949	47.31

The worst results are underlined. Average rows are displayed in gray.

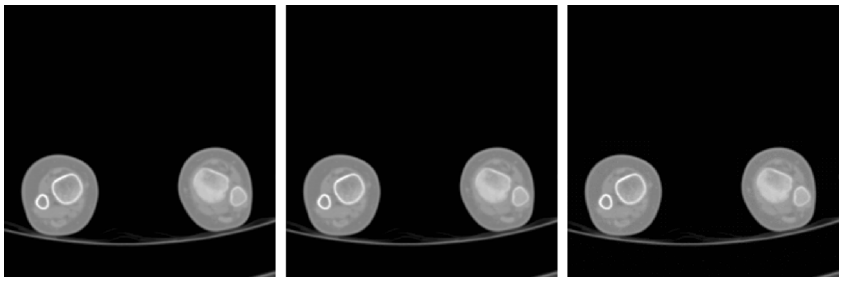

Fig. 7 Reconstructed images of CT_ Aperts. **a** Original. **b** PAE = 2 (PSNR = 62.21 dB). **c** PAE = 8 (PSNR = 49.81 dB)

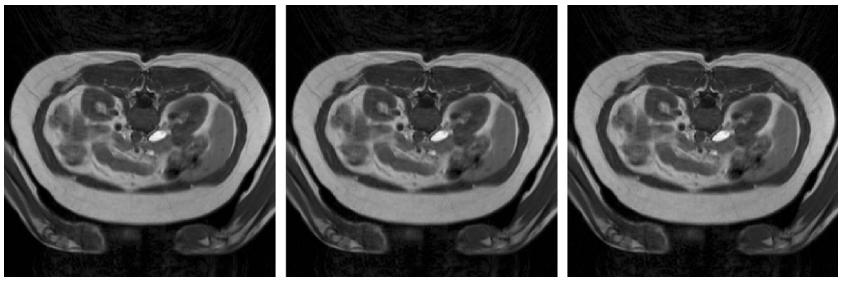

Fig. 8 Reconstructed images of MRI_liver_t1. **a** Original. **b** PAE = 2 (PSNR = 58.39 dB). **c** PAE = 8 (PSNR = 46.85 dB)

Experimental Results

Datasets Used for Experiments

We tested the proposed algorithm on 8-bit CT and MRI medical sequence images. More than 500 CT and 300 MRI slices from nine data sets were used. These data sets contain images from the Mallinckrodt Institute of Radiology Image Processing Laboratory, available at CIPR http://www.cipr.rpi.edu/resource/sequences/sequence01.html, and MRI images from the Computer Vision Group [23] at the University of Granada. Table 2 gives descriptions of these images. The first slices of each data set are shown in Fig. 5.

Analysis of Adaptive Block Size

The adaptive block size used in this paper is compared with the fixed block size in this subsection. Figure 6 shows the adaptive partitioning maps of two sample images. Because the block size used in our method is adaptively selected from among 32 × 32, 16 × 16, and 8 × 8, comparisons among these three conditions are made in Table 3. The adaptive block size outperforms the fixed block size because it can guarantee a more rational division of images and the improved use of local structures.

Analysis of Block Searching Method

The improved block searching method used in this paper is compared with the full search, as shown in Table 4. Using SAD characteristics during block searching can improve the encoding speed by 72–86 %. This makes up for the greater time consumed while providing the same search results as the full search.

Comparison of Near-Lossless Results

We use bit rate and PSNR to measure near-lossless compression performance. For many near-lossless methods, as PAE increases, PSNR decreases rapidly. Because medical images must maintain a high quality, PAE is always set to be small to meet the quality requirements of the reconstructed images. We use PAE = 2 to represent a small value and PAE = 8 to represent a relative large distortion.

We first compare the near-lossless compression performance of the proposed method with that of several state-of-the-art algorithms such as CALIC [24], JPEG-LS (version 2.1), JPEG2000 [25] (JasPer version 1.900.0), and SPIHT [26] (based on a 5/3 integer wavelet and 2 levels of transform). Table 5 shows a comparison of bit rates averaged across each data set and the PSNR results of our proposed method. The results of these state-of-the-art algorithms are lossless. In terms of bit rate, SPIHT performs poorly because its zerotree structure is more suitable for lossy compression. The proposed near-lossless method can efficiently compress medical images. When PAE is 2, the performance of our algorithm is comparatively equivalent to that of other algorithms. When PAE is 8, the compression performance of our algorithm is 16.6 %, 12.4 %, 21.5 %, and 48.7 % better than CALIC, JPEG-LS, JPEG2000, and SPIHT, respectively. For our proposed method, when PAE is 2, the average PSNR is approximately 59 dB, which is an excellent result; when PAE is 8, the average PSNR is approximately 47 dB, which is a good result. Figures 7 and 8 show reconstructed images under these two PAEs for two sample images. Visually, there are nearly no differences.

Table 6 provides a comparison of PSNR and bit rates for the proposed algorithm and some near-lossless algorithms under different PAEs. The bit rates of the proposed algorithm are similar or sometimes slightly lower than those of other

Table 6 Comparison of bit rates (in bpp) and PSNR (in dB) for the proposed method with some near-lossless methods (results are averaged across all data sets)

	PAE = 2		PAE = 8	
	PSNR	bpp	PSNR	bpp
DPCM [8]	45.21	2.30	38.30	1.40
EC [9]	49.23	1.03	38.88	0.30
JPEG-LS + WAT [10]	45.19	3.59	38.56	2.31
SPIHT [26]	45.21	2.30	40.17	1.20
SPIHT + AC [13]	45.17	2.12	38.17	0.79
BP coder [12]	44.33	4.16	36.34	1.92
Proposed	**59.21**	2.42	**47.31**	1.95

The averaged PSNR of the proposed algorithm are in bold

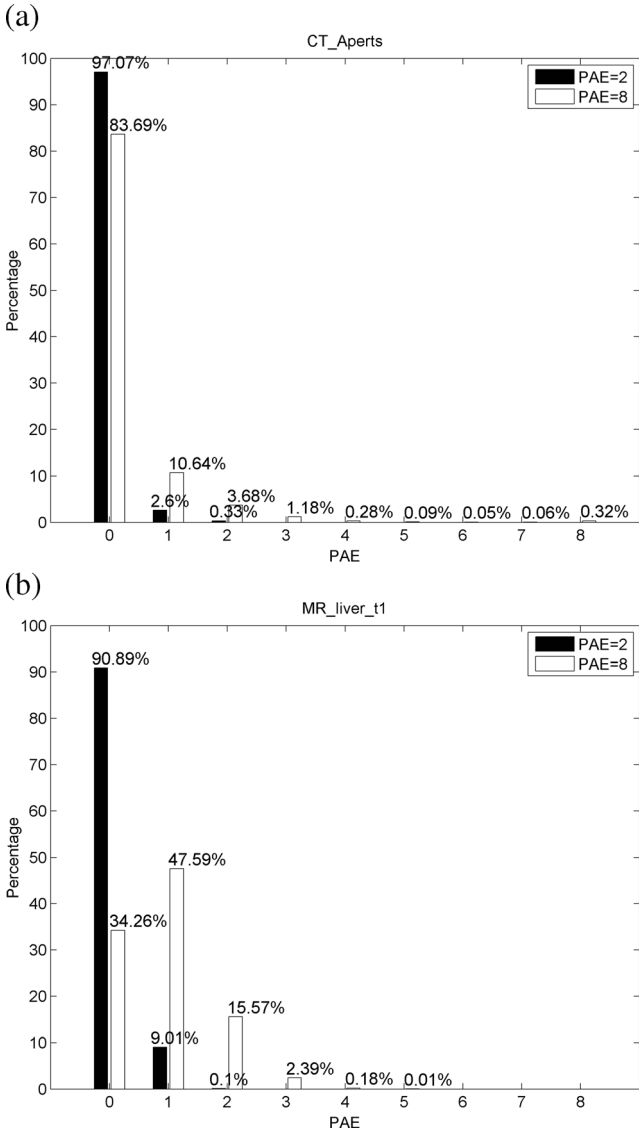

Fig. 9 Histogram of the percentage of image pixels with a certain distortion error

algorithms. The PSNR of our proposed algorithm is much more efficient than that of the comparison algorithms. The PSNR of our method with PAE = 8 is even better than the PSNR of other algorithms with PAE = 2. Due to the redundancy of Hadamard Transform coefficients, quantization is performed on the transformed coefficients rather than the prediction residuals, and after inverse transformation, errors are mainly found in the first coefficient of each 4 × 4 block, while the other coefficients are recovered nearly losslessly. Thus, the high quality of the reconstructed image is preserved even for a high PAE.

In Fig. 9, the percentages of image pixels with a certain distortion error with respect to the original image are reported for two sample images. Approximately 90 % of the image pixels are losslessly recovered, and the actual PAE may be lower than the theoretically obtained values of 2 and 8. For example, the second PAE in Fig. 9b is 5, rather than 8.

In conclusion, when compared with the state-of-the-art lossless algorithms, the near-lossless version of the proposed algorithm can achieve a higher compression ratio. When compared with other near-lossless algorithms, the proposed algorithm performs similarly in terms of compression ratio; meanwhile, it can provide much higher PSNR values, i.e., its reconstructed images are of better quality than those produced by other near-lossless algorithms.

Conclusions

Accounting for the characteristics of medical images, we propose a near-lossless compression algorithm for medical sequence images for possible diagnostic use with adaptive block size-based spatial prediction. The block-based prediction used breaks the neighborhood constraint for pixels, performing block matching directly in the spatial domain through block searching. An improved block searching method is proposed that improves searching speed. Before quantization, the prediction residuals are transformed by the Lossless Hadamard Transform to obtain high image quality. Our extensive experimental results demonstrate that the proposed adaptive prediction method is efficient for the near-lossless compression of medical images.

Acknowledgments This work was supported by the National Natural Science Foundation of China (grant nos. 61574102, 61404094, and 61204096), the Fundamental Research Fund for the Central Universities, Wuhan University (grant no 2042014kf0238), and the Hubei Province Science & Technology Pillar Program (grant no. 2015CFB536).

References

1. Placidi G: Adaptive compression algorithm from projections: Application on medical greyscale images. Comput Biol Med 39(11):993–999, 2009
2. Fidler A, Skaleric U, Likar B: The impact of image information on compressibility and degradation in medical image compression. Med Phys 33(8):2832–2838, 2006
3. Chen K, Ramabadran TV: Near-lossless compression of medical images through entropy-coded DPCM. IEEE Trans Med Imaging 13(3):538–548, 1994
4. Singh S, Kumar V, Verma HK: Adaptive threshold-based block classification in medical image compression for teleradiology. Comput Biol Med 37(6):811–819, 2007
5. Lee JO, Jang SK, Chen QS, et al: An efficient frame rate up-conversion method for mobile phone with projection functionality. IEEE Trans Consum Electron 53(4):1615–1621, 2007
6. Muhit AA, Pickering MR, Frater MR, et al: Video coding using fast geometry-adaptive partitioning and an elastic motion model. J Vis Commun Image Represent 23(1):31–41, 2012
7. Zhao XO, He ZH: Lossless image compression using super-spatial structure prediction. IEEE Signal Process Lett 17(4):383–386, 2010

8. Hartenstein H, Herz R, Saupe D: A comparative study of L1 distortion limited image compression algorithms. Proc Picture Coding Symp 51:55, 1997
9. Aràndiga F, Mulet P, Renau V: Lossless and near-lossless image compression based on multiresolution analysis. J Comput Appl Math 242:70–81, 2013
10. Caldelli R, Filippini F, Barni M: Joint near-lossless compression and watermarking of still images for authentication and tamper localization. Signal Process-Image Commun 21(10):890–903, 2006
11. ISO/IEC 14495–1: Information Technology-Lossless and Near-lossless Compression of Continuous Tone Still Images: Baseline. Dec. 1999. JPEG-LS source code available at: http://www.stat.columbia.edu/%7Ejakulin/jpeg-ls/mirror.htm, 1999
12. Miguel A, Riskin E, Ladner R, et al: Near-lossless and lossy compression of imaging spectrometer data: comparison of information extraction performance. SIViP 6(4):597–611, 2012
13. Yea S, Pearlman W: A wavelet-based two-stage near-lossless coder. IEEE Trans Image Process 15(11):3488–3500, 2006
14. Koga T: Motion-compensated interframe coding for video conferencing. Proc NTC'81. 1981: C9. 6.1-9.6.5
15. Ghanbari M: The cross-search algorithm for motion estimation. IEEE Trans Commun 38(7):950–953, 1990
16. Moshnyaga VG: A new computationally adaptive formulation of block-matching motion estimation. IEEE Trans Circuits Syst Video Technol 11(1):118–124, 2001
17. Lin Y, Zhuang Q, Yang R: Image reconstruction of dynamic MRI based on adaptive motion estimation. IEEE Int Conf ICARCV 1586–1590, 2012
18. Chen J, Zhou J, Yu S, et al: A Very Low Bit Rate Video Coding Combined with Fast Adaptive Block Size Motion Estimation and Nonuniform Scalar Quantization Multiwavelet Transform. Multimed Tools Appl 26(1):123–144, 2005
19. Wei ST, Tien CW, Liu BD, et al: Adaptive truncation algorithm for Hadamard-transformed H. 264/AVC lossless video coding. IEEE Trans Circuits Syst Video Technol 21(5):538–549, 2011
20. Lim C, Kim G, Yoon C, et al: Context modeling based lossless compression of radio-frequency data for software-based ultrasound beamforming. Biomed Signal Process 8(6):682–687, 2013
21. Sun J, Ren G, Wu Q: Image compression algorithm based on adaptive exp-Golomb coding. Opt Precis Eng 21(11):2973–2979, 2013
22. Nelson M, Gailly J L: The data compression book. New York: M&T Books, 1996, Chapter 6, order 3 arithmetic coding
23. Computer Vision Group [online]: Available at: http://decsai.ugr.es/cvg/index2.php. Accessed Aug 2011
24. Wu X, Memon N: CALIC—a context based adaptive lossless image codec. Proc ICASSP. 1996, 4: 1890–1893. CALIC source code available at: http://compression.graphicon.ru/download/sources/i_glless/codec.zip
25. ISO/IEC JTC 1/SC 29/WG 1, ISO/IEC FCD 15444–1, Information Technology – JPEG 2000 Image Coding System, 2000. JPEG2000 source code available at: http://www.kakadusoftware.com/
26. Said A, Pearlman W: A new, fast, and efficient image codec based on set partitioning in hierarchical trees. IEEE Trans Circuits Syst Video Technol 6(3):243–250, 1996