Tensor Networks for Image Compression

Author: Àlex Trujillo Boqué
*Facultat de Física, Universitat de Barcelona, Diagonal 645, 08028 Barcelona, Spain.**

Advisor: José I. Latorre

Abstract: Tensor Network methods are utilized in the development of an image compression algorithm. Visual information is represented via a Matrix Product State that undergoes a truncation process. Performance is compared with JPEG.

I. INTRODUCTION

Over the past decades, Tensor Networks (TN) have been used to efficiently represent and simulate entangled quantum many-body systems. States of such systems live in a Hilbert space that grows exponentially with the number of particles [1]. TN methods provide a representation of states that encodes quantum correlations between the n constituents of the system in a way that allows to reduce the parameters needed to poly(n) instead of exp(n).

An important family of TN are the Matrix Product States (MPS) [2], which arose from the Density Matrix Renormalization Group development [3]. They consist of a one-dimensional array of tensors, each for a site of the system, and encode correlations linearly along the sites. These sites can be *e. g.* spins, but can also be more abstract as they are in this project. The MPS states considered here have open boundary conditions, *i. e.* direct correlations between the first and the last site are not considered. Such a MPS can be written as

$$|\Psi\rangle = \sum_{\{\alpha_a\},\{i_a\}} \Gamma^{i_0}_{\alpha_0} \Gamma^{i_1}_{\alpha_0\alpha_1} \Gamma^{i_2}_{\alpha_1\alpha_2} \dots \Gamma^{i_{n-1}}_{\alpha_{n-2}} |i_0 \dots i_{n-1}\rangle, \quad (1)$$

with free indices i_a and ancillary or bond indices α_a for contraction. The i_a in the ket refer to qudits with p possible states. The Γ tensors can be seen as arrays composed of p matrices, except for the first and the last which contain vectors.

TN methods are useful in any system with direct product structure, not only quantum systems. The proposal here is to apply the TN methods to image compression.

Information stored in a block of pixels can be expressed at several levels of precision or detail if the image matrix is repeatedly coarse-grained in sub-blocks of increasing size. Coarse-graining provides a one-dimensional view along the depth of the image that can be expressed by ways of a MPS. As seen in Fig.(1), the levels of coarse-graining are identified as the sites of the system. To achieve this, an image, which is a matrix, is cast into a n-dimensional tensor of coefficients of a real ket, that defines a quantum state. This gives to the system a direct product structure. The quantum state is then represented in a MPS. Afterwards, a truncation scheme for the tensors with a maximum bond dimension for them is mandatory to achieve compression. This is the main idea developed in [4].

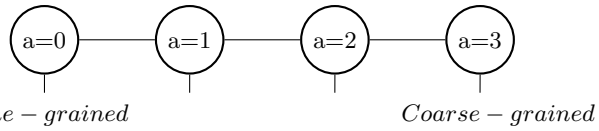

FIG. 1: A Matrix Product State with open boundary conditions in diagrammatic notation. The circles are its $n = 4$ sites, with a label a. Sites represent tensors and each belongs to a coarse-graining level. Tensor indices are the lines or bonds that emerge from the site. Bonds between sites indicate correlations between levels, and correspond to indices α_a shared by two adjacent tensors, *i. e.* contracted indices. Free bonds under the sites are open indices i_a with dimension p.

An algorithm is developed to implement this method along with some classical, well-known compression techniques. Its performance is benchmarked against JPEG.

II. THE ALGORITHM

The algorithm can be divided in five parts: an initial pre-processing, followed by a labelling of pixels and casting of the image into a real ket, which is then represented in terms of a MPS, a truncation and quantization of the MPS tensors, and finally entropy coding. Additionally, the decompression algorithm is exposed.

A. Initial pre-processing

First of all, the image is divided in square blocks. The side length of a block must be a power of two.

As usual in data compression methods that focus on the human limitations in resolution of detail, it is essential to work in the frequency space. Therefore in each square block a 2D discrete cosine transform is performed.

B. Labelling and casting

A square block of pixels in the frequency domain as considered here has information of low frequencies in its

*Electronic address: atrujibo7.alumnes@ub.edu

upper-left corner, and of high frequencies in the opposite corner. This distribution works well with the following step, inspired in the Renormalization Group: coarse-graining levels are defined and given a label a.

The finest-graining level ($a = 0$) contains $p = m^2$ pixels in each square sub-block, with $m = 2, 3, \ldots$, and pixels in every sub-block are given a position label $i_{a=0} \in [0,\ p-1]$. The label refers to a computational basis state in a local Hilbert space of dimension p.

This is repeated for the coarser-grained levels, where each division of the sub-blocks contains more than one pixel, as seen in Fig.(2). In each level a, every pixel receives a label i_a, it indicates in which of the p divisions it is contained at that particular level. Thus, if the lateral size of the pixel block is L, every pixel is assigned a total of $n = \log_p(L^2)$ labels, with $n \in \mathbb{N}$.

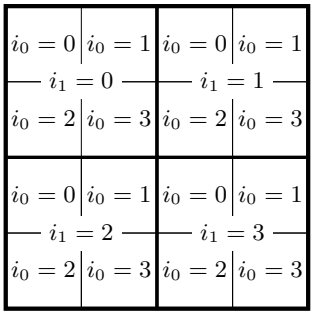

FIG. 2: Diagram of the labelling for a pixel block in the frequency space, with $L = 4$ and $p = 4$. Two divisions are performed ($n = 2$) and therefore i_a has two components. As pixel positions are in the frequency domain, the $i_1 = 0$ square contains the low frequency pixels, which are the most relevant. The correlations between levels in the upper-left square shall therefore be retained with more precision than the rest in the lossy steps of the algorithm.

Thus the size of the pixel blocks is related to the amount of correlations that will be encoded. Considering that a minimum of two levels are required to have correlations, $n > 1$. Moreover, p is a power of a natural number. This is a condition for the values of L allowed.

The previous labelling process can be interpreted as a mapping from the position of a pixel in the frequency space (x,y) to a n-dimensional array of indices, i_a:

$$i_a = \left\lfloor \frac{x \bmod (p^{\frac{a+1}{2}})}{p^{\frac{a}{2}}} \right\rfloor + \sqrt{p} \left\lfloor \frac{y \bmod (p^{\frac{a+1}{2}})}{p^{\frac{a}{2}}} \right\rfloor. \qquad (2)$$

Note the use of $\lfloor\ \rfloor$, the floor operator. To step into the quantum representation, a qudit of dimension p, represented by the ket $|i_a\rangle$, is assigned to each coarse-graining level. Qudits have the usual computational basis states $\{|0\rangle,\ldots,|p-1\rangle\}$. Consequently a multiple-qudit state is obtained, and represented by the ket $|i_0 \ldots i_{n-1}\rangle$, with the direct product. This state is the superposition of pure states with all possible combinations of values of i_a, and by construction each of this combinations corresponds to a pixel. For instance, the upper-left pixel of the image, which is labelled $i_a = (0, \ldots 0)$, gives its value to the real coefficient $c_{0\ldots0}$ that belongs to the pure state $|0 \ldots 0\rangle$. For an 8-bit grayscale image this value shall be an integer in the range [0, 255].

This identification is referred to as casting of the image into a real ket. Note that $c_{\{i_a\}} \in \mathbb{R}^n$. The resulting quantum state is then normalized. It will be entangled in general, unless the image has no texture at all.

$$|\Psi\rangle = \sum_{\{i_a\}}^{4} c_{i_0 \ldots i_{n-1}} |i_0 \ldots i_{n-1}\rangle. \qquad (3)$$

In a 16x16 pixel block $n = 4$ levels can be considered with $p = 4$. The exponential dimension of the Hilbert space in the direct product structure is manifest, since there are $p^n = 256$ pixels, identified as the coefficients $c_{\{i_a\}}$ of the ket. The current superposition of pure states is a representation of the quantum state that makes use of a tensor of coefficients with n indexes.

C. Matrix Product State representation

The objective in this step is to decompose the tensor of coefficients into n tensors with 2 or 3 indeces each.

$$c_{i_0 \ldots i_{n-1}} = \sum_{\{i_a\}}^{p} \sum_{\{\alpha_a\}}^{\{\chi_a - 1\}} \Gamma^{i_0}_{\alpha_0} \Gamma^{i_1}_{\alpha_0 \alpha_1} \Gamma^{i_2}_{\alpha_1 \alpha_2} \ldots \Gamma^{i_{n-2}}_{\alpha_{n-3} \alpha_{n-2}} \Gamma^{i_{n-1}}_{\alpha_{n-2}}. \qquad (4)$$

The range of α_a indexes is known as Schmidt rank, χ_a, and quantifies the correlations between levels of coarse-graining. Larger χ indicates more entropic correlations.

The procedure followed to obtain the $\Gamma^{i_a}_{\alpha_a \alpha_{a+1}}$ tensors is explained in [5]. It is in essence a succession of Schmidt decompositions, which are singular value decompositions (SVD) in the context of a quantum state.

First, the $p^n \times p^n$ density matrix ρ must be obtained. Next, the decomposition starts with a partial trace with respect to the levels $a = 1, \ldots n-1$ to obtain the reduced density matrix of the level $a = 0$.

$$Tr_{1 \ldots n-1}(\rho) = \rho_0 =$$

$$= \sum_{i_1 \ldots i_{n-1}} \begin{pmatrix} c^2_{0 i_1 .. i_{n-1}} & \cdots & c_{0 i_1 .. i_{n-1}} c_{3 i_1 .. i_{n-1}} \\ \vdots & \ddots & \vdots \\ c_{3 i_1 .. i_{n-1}} c_{0 i_1 .. i_{n-1}} & \cdots & c^2_{3 i_1 .. i_{n-1}} \end{pmatrix}. \qquad (5)$$

Similar operations are performed to obtain the reduced density matrices $\rho_{1 \ldots n-1}$, $\rho_{2 \ldots n-1}$ and so on until ρ_{n-1}. Summations of coefficients in the partial traces are always over the indices that do not appear in the subscript. The eigenvectors of such matrices are known as the Schmidt

vectors (SV) of the corresponding bi-partition, denoted by $|\psi^0\rangle$ for $a = 0$. Its eigenvalues are the square of the Schmidt coefficients (SC) $\lambda_{\alpha's}$. The SC's coincide for each pair of complementary bi-partitions, *e. g.* $|\psi^0\rangle$ and $|\psi^{1\ldots n-1}\rangle$, since they depend on mutual correlations. It is therefore natural that the bond indices α are associated to each set of SC and SV.

The first division results in the following Schmidt decomposed state, from the SV and SC of ρ_0 and $\rho_{1\ldots n-1}$,

$$|\Psi\rangle = \sum_{\alpha_0=0}^{\chi_0-1} \lambda_{\alpha_0} |\psi^0_{\alpha_0}\rangle \otimes |\psi^{1\ldots n-1}_{\alpha_0}\rangle, \tag{6}$$

SC are ordered in descending value. This is important for the correct distribution of information in the MPS. The $\otimes$ symbol is generally omitted below.

For every possible division the amount of non-zero SC is $\chi_a = p^b$, where b is the number of levels in the smallest bi-partition. However, if levels a and $a+1$ present a common pattern, their correlations are less entropic, consequently the quantum state is less entangled and $\chi_a < p^b$. Nonetheless, this is not the case for common images.

The SV's from Eq. (6), $|\psi^0_{\alpha_0}\rangle$, are expressed in terms of the computational basis of qudits, with a basis change matrix $\Gamma'^{i_0}_{\alpha_0}$. This matrix and the SC λ_{α_0} are contracted and so the first of the Γ tensors is obtained:

$$|\Psi\rangle = \sum_{i_0,\alpha_0} \lambda_{\alpha_0} \Gamma'^{i_0}_{\alpha_0} |i_0\rangle |\psi^{1\ldots n-1}_{\alpha_0}\rangle = \sum_{i_0,\alpha_0} \Gamma^{i_0}_{\alpha_0} |i_0\rangle |\psi^{1\ldots n-1}_{\alpha_0}\rangle, \tag{7}$$

Each SV $|\psi^{1\ldots n-1}_{\alpha_0}\rangle$ is expanded in terms of the computational basis of $|i_1\rangle$ and of auxiliary vectors $|\tau^{2\ldots n-1}_{\alpha_0,i_1}\rangle$,

$$\begin{aligned} |\Psi\rangle &= \sum_{\alpha_0,i_0} \Gamma^{i_0}_{\alpha_0} |i_0\rangle \sum_{i_1} |i_1\rangle |\tau^{2\ldots n-1}_{\alpha_0,i_1}\rangle = \\ &= \sum_{\alpha_0,i_0,i_1} \Gamma^{i_0}_{\alpha_0} |i_0 i_1\rangle |\tau^{2\ldots n-1}_{\alpha_0,i_1}\rangle, \end{aligned} \tag{8}$$

$|\tau^{2\ldots n-1}_{\alpha_0,i_1}\rangle$'s are then expressed in terms of the SV $|\psi^{2\ldots n-1}_{\alpha_1}\rangle$, eigenvectors of $\rho_{2\ldots n-1}$. This results in a basis change matrix with indices i_1 and α_1, for each α_0. Together they provide the next tensor $\Gamma^{i_1}_{\alpha_0,\alpha_1}$.

$$|\tau^{2\ldots n-1}_{\alpha_0,i_1}\rangle = \sum_{\alpha_1=0}^{\chi_1-1} \Gamma^{i_1}_{\alpha_0,\alpha_1} |\psi^{2\ldots n-1}_{\alpha_1}\rangle \tag{9}$$

$$|\Psi\rangle = \sum_{\alpha_0,\alpha_1,i_0,i_1} \Gamma^{i_0}_{\alpha_0} \Gamma^{i_1}_{\alpha_0,\alpha_1} |i_0 i_1\rangle |\psi^{2\ldots n-1}_{\alpha_1}\rangle \tag{10}$$

Successive decompositions as in Eq. (7-10) finally yield $|\psi^{n-1}_{\alpha_{n-2}}\rangle$. It is then expressed in the computational basis:

$$\begin{aligned} |\Psi\rangle &= \sum_{i_0\ldots i_{n-2}}^{p} \sum_{\{\alpha_a\}}^{\chi_a-1} \Gamma^{i_0}_{\alpha_0} \ldots \Gamma^{i_{n-2}}_{\alpha_{n-3},\alpha_{n-2}} |i_0 \ldots i_{n-2}\rangle |\psi^{n-1}_{\alpha_{n-2}}\rangle \\ &= \sum_{\{i_a\}}^{p} \sum_{\{\alpha_a\}}^{\chi_a-1} \Gamma^{i_0}_{\alpha_0} \Gamma^{i_1}_{\alpha_0\alpha_1} \ldots \Gamma^{i_{n-1}}_{\alpha_{n-2}} |i_0 \ldots i_{n-1}\rangle \end{aligned} \tag{11}$$

An expression like Eq. (1) is obtained. These tensors are an *exact* representation of the image.

The value of χ_a has its maximum $\chi_{max} = p^{\lfloor n/2 \rfloor}$ in the middle of the tensor array. In this representation of the image, $n \times p \times \chi^2$ parameters are needed instead of p^n. But χ's upper bound is O(exp(n)). Therefore, to make this representation efficient, the bond indices α_a are bound with a χ_{trunc}. The amount of parameters is then poly(n).

D. Truncation scheme

The truncation of indices can be approached from multiple points of view [4]. Here it is approached as follows: since the essential information is stored in the SV's associated to the highest SC's and they have been sorted by decreasing value, a χ_{trunc} is established for all α's. The less significant part of the tensors is not retained.

The optimal χ_{trunc} will depend on the textures of the image, as well as on the tolerated quality loss and the data compression ratio aimed to attain. However, $\chi_{trunc} > 1$ is necessary to have an entangled state and retain correlations in the truncated set $\{\Gamma_{trunc}\}$.

Once the truncation of the α indexes is performed, the set $\{\Gamma_{trunc}\}$ contains $n \times p \times \chi^2_{trunc}$ real numbers at most, which is poly(n). This is relatively efficient, but the compression would still not be competitive.

Integer truncation is applied to the real values in $\{\Gamma_{trunc}\}$ after they are scaled to the range [-128, 127]. Now each value in the MPS could be stored in a byte.

The following step is based in the quantization tables that were developed for the JPEG standard [6]. Not all numbers stored in the MPS are equally important, hence the interest in a custom precision for each established through quantization. This is implemented with integer division of every tensor by tables in the range [1, 127]. The less significant a value is, the larger its divisor. The resulting quantized tensors have many repeated values.

In the case of a JPEG table, values are distributed in a 2D matrix in the frequency space, therefore important values are those in the low frequency area. In the MPS tensors, *a priori* one only has slight clues of where the most significant values are. Plus, tables are not 2D, as they must be of the same shape as the MPS tensor array.

A first approach to find the adequate MPS quantization tables is to assign more precision to low α values since they come from a dominant SC. And to i_{n-1}=0, where low frequency values were in the coarsest-grained division. This leads to a set of pre-built tables.

After quantization, $\{\Gamma_{trunc}\}$ is stored, and the whole process is repeated for the remaining blocks in the image.

A genetic algorithm has been used to find nearly-optimal quantization tables, from the pre-built ones. Using this algorithm, changes in the tables gradually increase the quality and data compression ratio achieved, for a chosen set of parameters. The quantization tables are optimized for the full image, not for each pixel block.

E. Entropy coding

The final part of the compression is lossless entropy coding with Huffman encoding. All $\{\Gamma_{trunc}\}$ sets are joined in a string and encoded. Quantization tables are encoded as well. They are common for all sets and therefore suppose a small memory overhead. The Huffman tree adds a little overhead too.

The power of the Huffman encoding is greatly improved when most of the values in the $\{\Gamma_{trunc}\}$ are repeated. This is the advantage of quantization tables, as quantized tensors have many repeated values.

F. Decompression algorithm

To retrieve the image, first the compressed string is Huffman-decoded. Afterwards each $\{\Gamma_{trunc}\}$ set is multiplied by the quantization tables to reverse the quantization. The coefficients $c_{\{i_a\}}$ are recovered with a contraction of the α indices of the MPS and a denormalization,

$$pixel\{i_a\} = \sum_{\{\alpha_a\}} \Gamma^{i_0}_{\alpha_0} \Gamma^{i_1}_{\alpha_0\alpha_1} \dots \Gamma^{i_{n-1}}_{\alpha_{n-2}}, \tag{12}$$

and the pixel blocks are rebuilt with a mapping that is the inverse of Eq. (2). After that the discrete cosine transform is inverted.

As in the compression, the process is repeated for each block, and the image is reassembled.

III. RESULTS

The free parameters of the compression are:

- χ_{trunc}, the maximum value for the α indices.
- n, the number of coarse-graining levels considered.
- p, the number of divisions in each level.

Tests are conducted with $\chi_{trunc} \in [2,\ 8]$, $n = 4$ and $p = 4$. Performance of the compression algorithm is analyzed step by step, namely after:

- Representation in terms of the MPS with χ_{trunc}.
- Addition of integer truncation.
- With Huffman encoding but without quantization.
- Complete algorithm with quantization tables.

The bits needed to represent the image, with no overhead, are analytically obtained for the first and second steps:

$$N_{bits} = bit\ precision \cdot N_{blocks} \cdot (2p \cdot min(p, \chi_{trunc}) + \\ + \sum_{a=0}^{n-3} p \cdot min(\chi_a, \chi_{trunc}) \cdot min(\chi_{a+1}, \chi_{trunc})), \tag{13}$$

where $N_{blocks} = N_{pixels}/L^2$. Bit precision is taken as 64 bit for real values and 8 bit for integers. Therefore the difference between the size of the compressed image in steps one and two is an 8 factor. Data Compression Ratio (DCR) is then calculated as:

$$DCR = \frac{N_{pixels} \cdot bit\ precision}{N_{bits}} \tag{14}$$

The quality of the recovered image after decompression is measured with the Structural Similarity Index (SSIM). It takes values from -1 to 1, 1 is for identical images. An acceptable decompressed image should obtain at least $SSIM = 0.8$. High DCR with little SSIM loss is pursued.

Compression ratio and quality are calculated for two 8-bit grayscale images after the first three steps of the above list, for different χ_{trunc}, and shown in Table I.

8-bit gray, 512x512 px, Uncompressed size= $21.0 \cdot 10^5$ bits

	χ_{trunc}	2	3	4	8
No int. trunc.	DCR	0.67	0.33	0.20	0.11
	SSIM	0.8828	0.9388	0.9720	0.9935
Int. trunc.	DCR	5.33	2.67	1.60	0.89
	SSIM	0.8798	0.9339	0.9653	0.9845
Huffman enc.	DCR	7.39	3.54	2.10	1.15

8-bit gray, 256x256 px, Uncompressed size= $5.2 \cdot 10^5$ bits

No int. trunc.	DCR	0.67	0.33	0.20	0.11
	SSIM	0.8633	0.9291	0.9683	0.9945
Int. trunc.	DCR	5.33	2.67	1.60	0.89
	SSIM	0.8607	0.9249	0.9622	0.9861
Huffman enc.	DCR	7.25	3.50	2.10	1.15

TABLE I: Accumulative performance, indicated with data compression ratio and quality of the recovered image, and assessed after each of 3 non-definitive steps in the algorithm: before and after integer truncation, and after Huffman encoding. No overhead is considered in DCR. The quality measure used is SSIM, after Huffman encoding it is not indicated as it is a lossless compression step and quality remains the same. It is seen that integer truncation is necessary to attain compression, with minimal quality loss. DCR is independent of image size before Huffman encoding.

For the final step, performance is benchmarked against JPEG in Table II. Results can vary with the same χ_{trunc} now, as they depend on the tables used. JPEG performance is controlled with a quality parameter $Q \in [1,\ 99]$.

The two MPS-compressed 512x512 pixel images from Table II are shown in Fig.(3), along with the original and one of the JPEG-compressed images for comparison.

In the tests conducted, the MPS algorithm achieves 38-67% of JPEG's compression ratio at the same quality without overhead considered. The highest percentages are for smaller images and specially for lower SSIM.

IV. CONCLUSIONS

Tensor Networks, and particularly Matrix Product States, prove to be suitable for the development of com-

	Lat. size	512	256	512	256	256
	SSIM	0.8311	0.8122	0.9014	0.8910	0,9400
MPS	χ_{trunc}	2	2	3	3	4
	DCR	17.97	19.60	7.64	7.93	3.82
JPEG	Q	10	11	24	32	70
	DCR	33.14	29.06	20.06	16.58	8.90

TABLE II: Data Compression Ratio comparison between MPS compression and JPEG, at several fixed SSIM values. JPEG DCR is obtained from resulting file size. MPS results are for p=4 and n=4. The lateral size in pixels of the two images used is indicated. JPEG performs 1,5-2,6 times better than MPS compression without overhead.

FIG. 3: Comparison of the original 512x512 8-bit grayscale image (upper-left) with images compressed with the MPS algorithm (upper-right, DCR=17.97, SSIM=0.8311, and lower-left, DCR=7.64, SSIM=0.9014) and JPEG (lower-right, DCR=33.14, SSIM=0.8311). Note how the compression artifacts are characteristic of each algorithm, even though the quality measure is the same.

pression algorithms. However, currently used compression methods perform significantly better than the algorithm developed. Hence, it is not a competitive alternative yet.

This work has been limited to the optimization for certain chosen parameters, but extensive testing and optimization can still lead to a more competitive result. Parameters n and p should be given higher values, which would increase the amount of correlations involved and thus allow greater compression. Dynamic truncation of α indices along the MPS tensor chain with a non-constant, polynomially growing χ_{trunc} might also optimize the storing of correlations.

It is found that quantization tables should be optimized separately for each image to achieve maximum performance. Different textures, size and compression parameters lead to distinct tables. Unfortunately, this demands a great computational time. A solution could be the creation of general tables to cover many combinations of reasonable parameters, ranges of sizes, and structural characteristics, without individual optimization.

Furthermore, the geometry of the correlation network can be altered as well. The immediate first step is to substitute open boundary conditions with periodic conditions. This is achieved with the addition of common indices to the first and last tensors. As a consequence, the tensors are in a ring instead of in a chain, and additional correlations are encoded between the finest and coarsest levels. Similarly, further bonds between levels could be considered to find more entangled representations of images. Families of TN other than MPS might be used for this purpose. As it has been pointed in previous works [7], the use of entangled states for the processing of structured pieces of information has strong advantages. Such pieces of information can be images, but the applications might be readily extended to other fields.

Acknowledgments

The author thanks his advisor J. I. Latorre for guidance, and his parents for their constant support.

[1] R. Orús. "A Practical Introduction to Tensor Networks: Matrix Product States and Projected Entangled Pair States". Ann. of Phys. **349**: 117-158 (2014).

[2] D. Pérez-García, F. Verstraete, M. M. Wolf, J. I. Cirac. "Matrix Product State Representations". Quantum Inf. Comput. **7**, 401 (2007) arXiv:quant-ph/0608197v2

[3] S. R. White. "Density Matrix Formulation for Quantum Renormalization Groups". Phys. Rev. Lett. **69**, 2863 (1992)

[4] J. I. Latorre. "Image compression and entanglement". arXiv:quant-ph/0510031 (2005).

[5] G. Vidal. "Efficient Classical Simulation of Slightly Entangled Quantum Computations". Phys. Rev. Lett. **91**, 147902 (2003).

[6] G. K. Wallace. "The JPEG still picture compression standard". IEEE Trans. Cons. Electronics, **38** No.1 (1992)

[7] S. E. Venegas-Andraca, J. L. Ball. "Processing images in entangled quantum systems". Quantum Inf. Proc. **9**, 1-11 (2010)