DICTIONARY LEARNING-BASED IMAGE COMPRESSION

Hao Wang[1] *Yong Xia*[1*] *Zhiyong Wang*[2]

[1]Shaanxi Key Lab of Speech & Image Information Processing (SAIIP),
School of Computer Science, Northwestern Polytechnical University, Xi'an 710072, China
[2]Biomedical and Multimedia Information Technology (BMIT) Research Group,
School of Information Technologies, University of Sydney, NSW 2006, Australia

ABSTRACT

Dictionary learning based image compression has attracted a lot of research efforts due to the inherent sparsity of image contents. Most algorithms in the literature, however, suffer from two drawbacks. First, the atoms selected for image patch reconstruction scatter over the entire dictionary, which leads to a high coding cost. Second, the sparse representation of image patches is performed independently from the quantization of sparse coefficients, which may result in a sub-optimal solution. In this paper, we propose the entropy based orthogonal matching pursuit (EOMP) algorithm and quantization KSVD (QKSVD) algorithm for dictionary learning-based image compression. An entropy regularization term is utilized in EOMP to restrict atom selection, and hence reduces the coding cost, and an adaptive quantization method is incorporated into the dictionary learning procedure in QKSVD to minimize the reconstruction error and quantization error simultaneously. Experimental results on 10 standard benchmark images demonstrate that our proposed approach achieves better performance than several state-of-the-art ones at low bit rate, such as KSVD based compression approach, JPEG, and JPEG-2000.

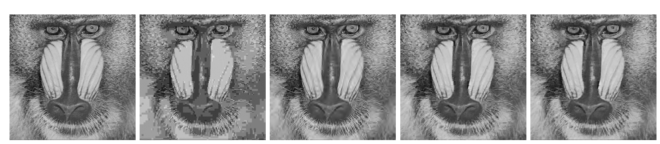
***Index Terms*—** Image compression, dictionary learning, adaptive quantization, information entropy

1. INTRODUCTION

With the advancements of imaging and computing technologies, imagery data have been increasingly produced and exchanged, such as the enormous number of images on various social media platforms. Although data storage has been more and more affordable, image compression still plays an important role in many computing systems in order to keep up with the pace between content size and storage capacity [1, 2, 3, 4].

In recent year, dictionary learning based compression algorithms have been proposed to further improve the compression ratio, due to the inherent sparsity of image contents [5, 6, 7]. Generally, there are four components in these algorithms: (1) separating an image into multiple patches, (2) sparsely reconstructing each patch with an over-completed dictionary, (3) quantizing the obtained sparse reconstruction coefficients, and (4) coding the quantized coefficients into a bitstream. Bryt et al. [8] applied the KSVD algorithm [9] to facial image compression and obtained a better performance than the JPEG-2000 standard [1] when the bit rate is very low. Zepeda et al. [10] proposed the iteration-tuned and aligned dictionary (ITAD) algorithm and applied it to facial image compression, too. By learning a general dictionary, some other dictionary learning-based algorithms were developed for general image compression [11, 12, 13, 14], which have a similar performance to JPEG-2000. Despite their success, these methods mainly focus on learning a better dictionary, but devoting less efforts to quantization and coding, which also have a profound impact on the compression ratio and decompression quality.

In this paper, a novel dictionary learning-based image compression approach is proposed, which is able to generate a universal dictionary that can benefit both quantization and coding. Specifically, the entropy based orthogonal matching pursuit (EOMP) algorithm is developed to keep the cost of coding at a low level, and the quantization KSVD (QKSVD) algorithm is designed to learn an adaptive quantization table during the dictionary learning process which is expected to be capable of quantizing the sparse reconstruction coefficients so as to guarantee a relatively high compressed image quality. The proposed approach has been evaluated against the KSVD algorithm, JPEG and JPEG-2000 image compression standards on ten benchmark images.



(a) (b) (c) (d) (e)

Fig. 1. (a) Original test image *Baboon* and its compressed versions generated by (b) JPEG, (c) JPEG-2000, (d) KSVD and (e) the proposed algorithm.

(a) (b) (c) (d) (e)

Fig. 2. Enlarged center part of (a) the image *Baboon* and its compressed versions generated by (b) JPEG, (c) JPEG-2000, (d) KSVD and (e) the proposed algorithm.

2. IMAGE COMPRESSION ALGORITHM

The proposed approach learns a universal dictionary and a quantization table using a relatively large set of training images at the following four steps: (1) dividing each training image into partly overlapped patches in the raster-scan order with a stride of 2 in both directions to explore all potential patch patterns; (2) applying the self-organizing mapping (SOM) [15, 16] algorithm to a randomly selected subset of mean-subtracted patches to generate approximate cluster-s, which are then used to initialize the K-means algorithm for grouping all mean-subtracted patches into those clusters; (3) applying the proposed EOMP and QKSVD algorithms to learn the corresponding dictionaries and quantization tables to those clusters; and (4) concatenating all learned dictionaries and quantization tables into a universal dictionary and a merged quantization table respectively, which are stored at both the encoder and decoder.

Image compression also consists of four steps: (1) dividing an image into patches of the same size to training patches without overlap; (2) sparsely reconstructing each mean-subtracted patch on the universal dictionary using the EOM-P algorithm; (3) quantizing the sparse reconstruction coefficients using the learned quantization table; and (4) encoding those mean values of patches and quantized coefficients using the run-length encoding (RLE) [17] and the huffman coding [18]. As an inverse process of compression, image decompression can be achieved by multiplying the coefficient matrix and the universal dictionary, followed by concatenating obtained patches into an image.

2.1. EOMP Algorithm

In dictionary learning-based image compression, the distribution of selected atoms has a great impact on the compression ratio. For instance, there are two signals $\boldsymbol{s}_i$ and $\boldsymbol{s}_j$. Signal $\boldsymbol{s}_i$ can be sparsely reconstructed using the atoms $\boldsymbol{d}_i$ and $\boldsymbol{d}_j$, $\boldsymbol{s}_j$ can be reconstructed using the atoms $\boldsymbol{d}_p$ and $\boldsymbol{d}_q$. Nevertheless, since the dictionary $\boldsymbol{D}$ is over-completed, it might be possible to reconstruct signal $\boldsymbol{s}_j$ with a similar accuracy using the atoms $\boldsymbol{d}_j$ and $\boldsymbol{d}_p$. Thus, if we choose the former reconstruction scheme, we have to encode the indices of four atoms; whereas if we choose the latter, we only need to encode the indices of three atoms. However, the conventional OMP [19] does not take the distribution of selected atoms into consideration. To overcome this drawback, we incorporate an entropy term, which poses a constraint to the selection of atoms, into the following objective function of sparse reconstruction

$$\begin{array}{c} \hat{\boldsymbol{A}} = \arg\min_{\boldsymbol{A}} \left\{ \|\boldsymbol{S} - \boldsymbol{D}\boldsymbol{A}\|_F^2 - \eta \boldsymbol{p}^T \log \boldsymbol{p} \right\} \\ s.t.\ \|\boldsymbol{a}_{\cdot j}\|_0 \leqslant k_{max}, 1 \leqslant j \leqslant K_n \end{array}, \tag{1}$$

where $\boldsymbol{S}$ is the ensemble of mean-subtracted patches, $\boldsymbol{D} \in \mathbb{R}^{K_m \times K_d}$ is the dictionary, $\boldsymbol{A} \in \mathbb{R}^{K_d \times K_n}$ is the ensemble of sparse reconstruction coefficients, $\boldsymbol{a}_{\cdot j}$ is the j^{th} column in $\boldsymbol{A}$, K_d is the number of atoms, k_{max} is the sparsity constraint, and $\boldsymbol{p} = (p_1, p_2, \ldots, p_{K_n})^T$ is a probability vector with each p_i representing the probability of selecting atom $\boldsymbol{d}_i$. We can approximate the probability $\boldsymbol{p}_i$ as follows

$$p_i = \frac{\|\boldsymbol{a}_{i\cdot}\|_0}{\|\boldsymbol{A}\|_0 + \delta}, \tag{2}$$

where $\boldsymbol{a}_{i\cdot}$ implies i^{th} row in $\boldsymbol{A}$, δ is a small value. The sparse representation problem given in Eq. 2 can be solved by the EOMP algorithm described in Algorithm 1.

Algorithm 1 The EOMP Algorithm

Input: Signal matrix $\boldsymbol{S} = (\boldsymbol{s}_1, \ldots, \boldsymbol{s}_{K_n})$, over-complete dictionary $\boldsymbol{D}$, sparsity prior k_{max}, reconstruction error ϵ, probability vector $\boldsymbol{p}$.

Output: Sparse representation matrix $\boldsymbol{A}$, probability vector $\boldsymbol{p}$.

Initialize $\boldsymbol{A} = 0$
for $n = 1$ to K_n **do**
  Initialize $i = 0, k^{(0)} = 0, \boldsymbol{r}^{(0)} = \boldsymbol{0}, \mathcal{S}^{(0)} = \varnothing$
  while $(k^{(i)} < k_{max}) or (\|\boldsymbol{r}^{(i)}\|_2^2 > \epsilon^2)$ **do**
    $i = i + 1$
    Find an atom which minimizes the cost function, $\forall \boldsymbol{d}_{\cdot j}, \hat{\boldsymbol{d}} = \arg\min_{\boldsymbol{d}_{\cdot j}, a_j} \left\{ \|a_j \boldsymbol{d}_{\cdot j} - \boldsymbol{r}^{(\boldsymbol{i}-\boldsymbol{1})}\|_2^2 - \eta p_j \log p_j \right\}$
    Put $\hat{\boldsymbol{d}}$ into support set, $\forall \hat{\boldsymbol{d}} \notin \mathcal{S}^{(i-1)}, \mathcal{S}^{(i)} = \mathcal{S}^{(i-1)} \cup \{\hat{\boldsymbol{d}}\}$
    $k^{(i)} = k^{(i-1)} + 1$
    Update $\boldsymbol{a}_{\cdot n}^{(i)}$ to minimize reconstruction error with atoms in $\mathcal{S}^{(i)}$
    Update residual $\boldsymbol{r}^{(i)} = \boldsymbol{s}_n - \boldsymbol{D}\boldsymbol{a}_{\cdot n}^{(i)}$
  end while
end for
Calculate probability vector $\boldsymbol{p}$, $p_i = \frac{\|\boldsymbol{a}_{i\cdot}\|_0}{\|\boldsymbol{A}\|_0 + \delta}$

2.2. QKSVD Algorithm

The obtained sparse reconstruction coefficients matrix $\boldsymbol{A}$ is decimal and need to be quantized in the following way

$$\begin{array}{c} \hat{\boldsymbol{D}} = \arg\min_{\boldsymbol{D}} \|\boldsymbol{S} - \boldsymbol{D} \cdot Q(\boldsymbol{A})\|_F^2 \\ s.t.\ \|\boldsymbol{a}_{\cdot j}\|_0 \leqslant k_{max}, 1 \leqslant j \leqslant K_n \end{array}, \tag{3}$$

where $Q(\cdot)$ is the quantization operator.

Let each element of the coefficient matrix $\boldsymbol{A}$ be denoted by a tuple (a_{dn}, idx_{dn}), where idx_{dn} is the position of a_{dn} in $\boldsymbol{A}$. We firstly sort all elements into an ordered sequence $L = \{a_1, a_2, \ldots, a_{K_d \times K_n}\}$, and then partition the sequence into k sub-sequences $\{L_1, L_2, \ldots, L_k\}$ by minimizing the following sum of square error (SSE).

$$\{\hat{L_1}, \hat{L_2}, \ldots, \hat{L_k}\} = \arg \min_{\{L_1,\ldots,L_k\}} \sum_{i=1}^{k} \sum_{j=1}^{|L_i|} (a_{i_j} - \bar{L_i})^2, \quad (4)$$

where a_{i_j} is the j^{th} element of L_i, and $\bar{L_i}$ is the mean of L_i. Since L is ordered, solving the partition problem in Eq. 4 is equivalent to identifying a set of $k-1$ cut-off points, denoted by C. Thus, a k-consecutive partition of L can be denoted by $\mathcal{P}(L, k, C)$

Lemma. *Let $\mathcal{P}(L,k,C)$ that minimizes SSE be defined as the minimal k-consecutive partition $\mathcal{P}_m(L,k,C)$. $\mathcal{P}(L-L_k, k-1, C-\{c_{k+1}\})$ must be the minimal $(k-1)$-consecutive partition of $L-L_k$, which implies*

$$\begin{aligned} &SSE(\mathcal{P}_m(L,k,C)) \\ =&SSE(\mathcal{P}_m(L-L_k, k-1, C-\{c_{k+1}\})) \\ &+\sum_{j=1}^{|L_k|}(a_{k_j} - \bar{L_k})^2 \end{aligned} . \quad (5)$$

Proof. Assume that $\mathcal{P}_m(L-L_k, k-1, C-\{c_{k+1}\})$ is not the minimal $(k-1)$-consecutive partition of $L-L_k$, then we can find the minimal $(k-1)$-consecutive partition $\mathcal{P}'_m(L-L_k, k-1, C-\{c_{k+1}\})$, thus $SSE(\mathcal{P}'_m(L-L_k, k-1, C-\{c_{k+1}\})) < SSE(\mathcal{P}_m(L-L_k, k-1, C-\{c_{k+1}\}))$. According to Eq. 5, $SSE(\mathcal{P}_m(L,k,C))$ can be smaller in this case. However, $SSE(\mathcal{P}_m(L,k,C))$ is the minimal k-consecutive partition of L. Hence, the assumption is incorrect. □

This lemma shows that the k-consecutive partition problem in Eq. 4 has optimal sub-structure and overlapping subproblems. Therefore, we can solve it efficiently using the dynamic programming algorithm [20].

To quantize larger coefficients with a larger step length and smaller ones with smaller step length, we use different step lengths among L_i's, but the same step length inside L_i. We employ a series of 8-bit binary codes $b_0b_1b_2b_3b_4b_5b_6b_7$ to represent different coefficients. For instance, if we partition the ordered sequence L into 8 parts, the structure of the code is: b_0 represents whether the coefficient is positive or negative, $b_1b_2b_3$ tells us which part L_i does the coefficient belong to, $b_4b_5b_6b_7$ is the dynamic range of the modulus of coefficients in part L_i. We embed the quantization in dictionary learning stage to construct a dictionary which is capable to minimize reconstruction error and quantization error at the same time.

Algorithm 2 The QKSVD Algorithm

Input: Signal matrix $\boldsymbol{S} = (\boldsymbol{s}_1, \ldots, \boldsymbol{s}_{K_n})$, reconstruction error ϵ.

Output: Learnt dictionary $\boldsymbol{D}$.

Initialize $i = 0$, probability vector $\boldsymbol{p} = \{\frac{1}{K_n}\}_{j=1}^{K_n}$

while $\|\boldsymbol{S} - \boldsymbol{D}^{(i)}\boldsymbol{A}^{(i)}\|_F^2 \geqslant \epsilon^2$ **do**

  $i = i + 1$

  Represent signal matrix $\boldsymbol{S}$ with the EOMP algorithm

  Quantize the sparse coefficient matrix $\hat{\boldsymbol{A}} = Q(\hat{\boldsymbol{A}})$

  for $j = 1$ to K_d **do**

    Record all signal vectors that use atom $\boldsymbol{d}_{\cdot j}$, $\Omega = \{q | a_{jq} \neq 0\}$

    Calculate error matrix $\boldsymbol{E} = \boldsymbol{S} - \sum_{k \neq j} \boldsymbol{d}_{\cdot k}\boldsymbol{a}_{k\cdot}$

    Calculate error sub-matrix $\boldsymbol{E}' = \boldsymbol{E}_{\cdot\Omega}$

    Decompose $\boldsymbol{E}'$ with SVD decomposition, $\boldsymbol{E}' = \boldsymbol{U\Delta V}^T$

    Update atom $\boldsymbol{d}_{\cdot j} = \boldsymbol{u}_{\cdot 1}$

    Update coefficient $\boldsymbol{a}_{j\Omega} = \Delta_{11}\boldsymbol{v}_{1\cdot}$

  end for

end while

3. EXPERIMENTS AND RESULTS

The proposed algorithm was compared against JPEG, JPEG-2000 and the KSVD algorithms on 10 benchmark images, including the *baboon, boat, cell, couple, elaine, lena, man, peppers, photography*, and *satellite*. In our algorithm, the universal dictionary and the adaptive quantization table are learned on 2525 training images randomly selected from the Caltech-101 dataset, the patch size was set to 16×16 with stride of 2, the dictionary was initialized with the DCT coefficients, the maximal number of iterations in QKSVD was set to 10, and weighting parameter η in Eq. 2 was set to 30. JPEG and JPEG-2000 were implemented by the MATLAB function $imwrite$ with default parameter setting, and KSVD was implemented according to [8]. To make a fair comparison, we took the huffman table into consideration when calculating the compression ratio of JPEG and JPEG-2000. The quality of compressed images was measured by the peak signal-to-noise ratio (PSNR).

Fig. 1 shows the test image *Baboon* and its compressed versions generated by four algorithms at a bit rate of 0.18 bpp. To highlight the difference of performance of four algorithms, the central part of this image was enlarged and displayed in Fig. 2. It reveals that the image compressed by JPEG suffers from severe blocking artefacts, the images compressed by JPEG-2000 and KSVD have blurred edges, and the compressed image produced by our algorithm preserves more visual details.

The PSNRs of the ten test images compressed at different bit rates were given in Table 1. It shows that JPEG-2000 and KSVD have a similar performance, and the proposed algo-

Table 1. PSNR of the ten images compressed by JPEG (top left), JPEG-2000 (top right), KSVD(bottom left) and the proposed algorithm (bottom right) at different bit rates.

Baboon	0.18bpp		0.26bpp		0.34bpp	
	20.18	22.78	21.84	23.76	23.02	24.80
	23.98	**26.07**	24.99	**26.46**	25.75	**26.95**
Boat	0.16bpp		0.23bpp		0.28bpp	
	25.12	29.32	27.67	**30.95**	29.25	**32.08**
	29.65	**29.90**	30.23	30.69	31.19	31.47
Cell	0.16bpp		0.21bpp		0.25bpp	
	27.31	**35.79**	31.13	**37.77**	33.07	**38.75**
	29.66	30.15	30.28	30.89	30.51	31.33
Couple	0.17bpp		0.23bpp		0.29bpp	
	25.44	30.27	27.96	31.71	29.86	32.85
	30.78	**31.09**	31.56	**32.24**	32.51	**33.11**
Elaine	0.17bpp		0.23bpp		0.28bpp	
	28.73	**33.96**	31.46	**34.85**	32.89	**35.47**
	32.85	32.99	33.66	33.80	34.19	34.20
Lena	0.16bpp		0.21bpp		0.25bpp	
	25.89	31.46	28.60	32.68	30.21	**33.84**
	31.73	**32.04**	32.63	**32.75**	32.43	32.59
Man	0.18bpp		0.25bpp		0.32bpp	
	23.88	26.73	25.96	28.06	27.29	29.14
	28.55	**29.58**	29.99	**30.34**	30.64	**30.86**
Peppers	0.16bpp		0.21bpp		0.25bpp	
	25.97	**32.71**	29.64	**34.19**	31.12	**35.11**
	31.79	31.93	31.96	32.81	32.19	32.88
Photo-graphy	0.11bpp		0.15bpp		0.18bpp	
	24.73	31.02	26.59	32.04	29.21	**33.92**
	31.16	**31.16**	31.98	**32.10**	32.63	32.65
Satellite	0.18bpp		0.25bpp		0.33bpp	
	23.67	**27.09**	26.11	**28.12**	27.13	**29.01**
	23.34	23.56	24.07	24.44	25.45	25.55

rithm achieves the highest PSNR on six out of ten test images when the bit rate is low.

4. DISCUSSIONS

4.1. Parameter Settings

The weighting parameter η in Eq. 3 represents the contribution of the entropy term, and hence plays an important role in learning the dictionary. To determine the best value of η, we performed the proposed algorithm with different values of η and plotted the average PSNR and the average compression ratio obtained on the training images in Fig. 3. It shows that both lines peak at $\eta = 30$, and hence we empirically set η to 30 in our experiment.

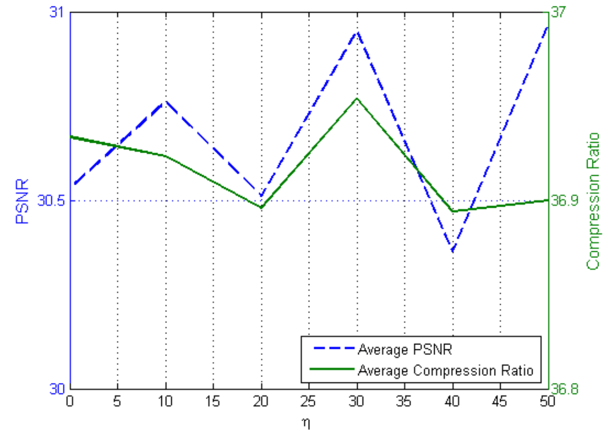


Fig. 3. Plot of average PSNR and average compression ratio versus the weighting parameter η.

4.2. Computational Complexity

Since we extracted patches of size 16×16 with stride of 2 on 2525 training images, we got more than 1.5 million patches. To reduce the time cost, we randomly selected 20% patches for dictionary learning, which is extremely time-consuming. Even in this case, it took more than 60 hours to obtain the universal dictionary (Intel Core(TM) i5-4460 3.20 GHz, 8 GB RAM and MATLAB R2014a). Fortunately, the offline dictionary learning needs to be performed only once.

The time complexity of the EOMP algorithm is $\mathcal{O}(K_n MN k_{max})$, where MN is the size of a patch. The time complexity of the huffman coding is $\mathcal{O}(W \log W)$, where W is the number of code words. Thus, the total time complexity of encoding is $\mathcal{O}(K_n MN k_{max} + W \log W)$. In practice, it took about 35 seconds on average to encode an image of size 512×512 at 0.18 bpp. Decoding is simply achieved by matrix multiplication, and thus can be done within 1 second.

5. CONCLUSION

In this paper, we present a novel dictionary learning-based image compression approach, which employs the newly designed EOMP and QKSVD algorithms. Our pilot results suggest that the proposed approach is able to achieve better image compression performance than the benchmark JPEG, JPEG-2000 and KSVD algorithms.

6. ACKNOWLEDGEMENTS

This work was supported in part by the National Natural Science Foundation of China under Grants 61471297, in part by the Natural Science Foundation of Shaanxi Province, China under Grant 2015JM6287, and in part by the Australian Research Council (ARC) Grants.

7. REFERENCES

[1] Athanassios Skodras, Charilaos Christopoulos, and Touradj Ebrahimi, "The JPEG 2000 still image compression standard," *IEEE Signal Processing Magazine*, vol. 18, no. 5, pp. 36 – 58, 2001.

[2] S. Milani, M. Tagliasacchi, and S. Tubaro, "Discriminating multiple JPEG compression using first digit features," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2014, pp. 2253–2256.

[3] K. Gregor, F. Besse, D. J. Rezende, I. Danihelka, and D. Wierstra, "Towards conceptual compression," in *Advances In Neural Information Processing Systems*, 2016, pp. 3549–3557.

[4] J. Ballé, V. Laparra, and E. P. Simoncelli, "End-to-end Optimized Image Compression," *ArXiv e-prints*, Nov. 2016, 1611.01704.

[5] W. Dong, L. Zhang, G. Shi, and X. Li, "Nonlocally centralized sparse representation for image restoration," *IEEE Transactions on Image Processing*, vol. 22, no. 4, pp. 1620–1630, April 2013.

[6] Li Xu, Shicheng Zheng, and Jiaya Jia, "Unnatural L0 sparse representation for natural image deblurring," in *The IEEE Conference on Computer Vision and Pattern Recognition*, June 2013.

[7] J. Zhang, D. Zhao, and W. Gao, "Group-based sparse representation for image restoration," *IEEE Transactions on Image Processing*, vol. 23, no. 8, pp. 3336–3351, Aug 2014.

[8] Ori Bryt and Michael Elad, "Compression of facial images using the K-SVD algorithm ," *Journal of Visual Communication & Image Representation*, vol. 19, no. 4, pp. 270–282, 2008.

[9] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.

[10] J. Zepeda, C. Guillemot, and E. Kijak, "Image compression using sparse representations and the iteration-tuned and aligned dictionary," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 5, pp. 1061–1073, 2011.

[11] K. Skretting and K. Engan, "Image compression using learned dictionaries by RLS-DLA and compared with k-svd," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2011, pp. 1517–1520.

[12] M. Xu, J. Lu, and W. Zhu, "Sparse representation of texture patches for low bit-rate image compression," in *2012 Visual Communications and Image Processing*, Nov 2012, pp. 1–6.

[13] M. Xu, S. Li, J. Lu, and W. Zhu, "Compressibility constrained sparse representation with learnt dictionary for low bit-rate image compression," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 10, pp. 1743–1757, Oct 2014.

[14] X. Zhang, W. Lin, S. Ma, S. Wang, and W. Gao, "Rate-distortion based sparse coding for image set compression," in *2015 Visual Communications and Image Processing*, Dec 2015, pp. 1–4.

[15] Juha Vesanto and Esa Alhoniemi, "Clustering of the self-organizing map," *IEEE Transactions on Neural Networks*, vol. 11, no. 3, pp. 586–600, 2000.

[16] Kohei Arai, "Image clustering method based on self organization mapping: SOM derived density maps and its application for landsat thematic mapper image," *International Journal of Advanced Research in Artificial Intelligence*, vol. 2, no. 5, 2013.

[17] Micha Stabno and Robert Wrembel, "RLH: Bitmap compression technique based on run-length and huffman encoding," *Information Systems*, vol. 34, no. 45, pp. 400 – 414, 2009.

[18] A. Jas, J. Ghosh-Dastidar, Mom Eng Ng, and N. A. Touba, "An efficient test vector compression scheme using selective huffman coding," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 22, no. 6, pp. 797–806, 2003.

[19] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition," in *Proceedings of 27th Asilomar Conference on Signals, Systems and Computers*, Nov 1993, pp. 40–44 vol.1.

[20] Richard E. Bellman, Stuart E. Dreyfus, Richard Bellman, Stuart Dreyfus, O. Gross, S. Dreyfus, M. Freimer, S. Dreyfus, Richard Bellman, and William Karush, *Applied Dynamic Programming:On a Transcendental Curve*, Princeton University Press, 2015.