

# Image Compression Approach using Segmentation and Total Variation Regularization

Ahmad Shahin, Walid Moudani, Fadi Chakik  
Doctoral School for Science and Technologies  
Lebanese University  
Tripoli, Lebanon

Received: March 10, 2021. Revised: April 12, 2021. Accepted: April 15, 2021. Published: April 28, 2021.

**Abstract**—In this paper we present a hybrid model for image compression based on segmentation and total variation regularization. The main motivation behind our approach is to offer decode image with immediate access to objects/features of interest. We are targeting high quality decoded image in order to be useful on smart devices, for analysis purpose, as well as for multimedia content-based description standards. The image is approximated as a set of uniform regions: The technique will assign well-defined members to homogenous regions in order to achieve image segmentation. The Adaptive fuzzy c-means (AFcM) is a guide to cluster image data. A second stage coding is applied using entropy coding to remove the whole image entropy redundancy. In the decompression phase, the reverse process is applied in which the decoded image suffers from missing details due to the coarse segmentation. For this reason, we suggest the application of total variation (TV) regularization, such as the Rudin-Osher-Fatemi (ROF) model, to enhance the quality of the coded image. Our experimental results had shown that ROF may increase the PSNR and hence offer better quality for a set of benchmark grayscale images.

**Keywords**—Image Compression; Segmentation; AFcM; Entropy Coding; ROF; Total Variation Regularization.

## I. INTRODUCTION (*Heading 1*)

Image compression deals with minimizing the amount of data needed to represent a digital image by removing redundant data. It involves encoding a 2-D array of pixels into a statistically uncorrelated data set. This transformation is applied before image storage or transmission. Afterward, the coded image is decompressed in order to reconstruct either the exact original image or an approximated version of the original one. For this reason, data compression system requires the definition of two functions: compression and decompression.

Most of the contemporary image coding techniques exploit the quantization principle, such as the JPEG standard based on the Discrete Cosine Transform (DCT) [1] or in the Log-exp transformation model [2]. The major limitation in JPEG is the blocky appearance of the decoded image. The Log-exp model suffers from its computation time which will be penalizing in real-time applications. The quantization stage is the core of

every lossy image encoding algorithm: in the encoder part, quantization means partitioning of the input data into a smaller set of values [3].

Increasingly, many studies had proved the involvement of Partial Differential Equation (PDE) and in particular Total Variation (TV) in the image processing field. They are largely the essential element of image compression and restoration [4, 14].

The basic idea of applying PDEs is to encode some image points (landmarks) selected according to many criteria such as edges or corners whereas in the decompression phase, the decoded image with incomplete information will be fulfilled by applying PDE's anisotropic diffusion [5]. Chan [13] proposed using PDE techniques in wavelet based image processing to remove noise and reduce edge artifacts generated by wavelet thresholding.

TV regularization offers algorithms for image processing in order to remove noise. It is based on regularizing criterion for solving inverse problems. It has proved to be quite efficient for regularizing images without smoothing the boundaries of the objects [15]. Takahiro [10] presented a restoration-type decoding approach based on the TV regularization that recovers a quality-improved image from the JPEG-compressed data, while not only suppressing the occurrence of the coding artifacts particular to the JPEG compression but also removing the camera's noise to some extent. TV denoising was also applied on SAR images in by decomposing the images into structure and texture [11, 12].

In this study, we suggest the exploitation of fuzzy clustering and TV regularization in order to achieve good image quality after compression with a possibility to define image features by the mean of clustering or segmentation. This work is an extension of the proposed approach illustrated in [6].

Image segmentation is an essential and important phase in image analysis in order to produce meaningful features of interest. The purpose behind this step is to formulate the image into regions or clusters. In our work, we chose the fuzzy logic and more precisely fuzzy sets-based algorithms to offer a clustering tool [7, 8, 9]. The fuzzy sets are a generalization of the classical set theory but they offer greater flexibility to

detect accurately the various aspects of vagueness or imperfectness in image information.

The remainder of this paper is structured as follows: Section 2 presents our system architecture and the description of the related theories such as fuzzy clustering and total variation principles. The description of these theories is useful to better understand the core of our work. In section 3, the assessment of our proposed approach is illustrated with the help of experimental results and comparisons. The paper concludes with section 4 by providing an overview of the contributions and some potential perspectives.

## II. SYSTEM ARCHITECTURE AND METHODS

The main purpose of this section is to explain the relevant theories which will be helpful for further understanding and discussion. Fig. 1 illustrates the system architecture and the principal steps performed in our proposed approach.

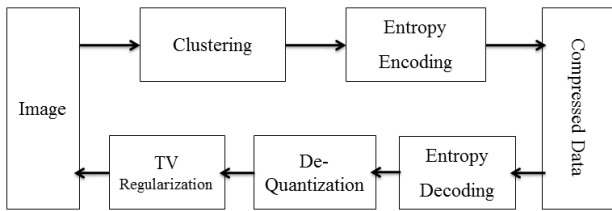


Fig. 1 The proposed image compression/decompression paths applying clustering and TV regularization

In our model, the compression algorithm is performed in the following way:

1. Firstly, unsupervised clustering algorithm is applied on the input image. We suggest the use the AFcM [9] as a tool to segment an image. At the end of this step, well-defined members are assigned to the cluster  $C$  with the highest membership.
2. A second stage of entropy encoding is executed and applied on image clusters. This will help reducing the large amount of redundancy. Multiple studies was performed and improved to find effective techniques in order to encode information without dropping their principal entities. Among several algorithms recently applied, we suggest the use of Huffman or Arithmetic coding algorithms.

Accordingly, the order of processing appears reversed in the corresponding decompression phase:

1. Apply entropy decoding.
2. De-quantization which assigns the well-defined members with the corresponding cluster center is carried out.
3. Apply TV regularization. We suggest the use of ROF model [14]. It is a method, most often used in image processing in order to remove noise. It offers a

regularizing criterion for solving inverse problems. It has proved to be quite efficient for regularizing images without smoothing the boundaries of the objects [15].

In the remaining part of this section, we will describe the methods used in the proposed compression/decompression algorithms.

### A. Image Segmentation

Image Segmentation (we may also refer to Clustering) is the process of organizing objects in groups having similar properties. Clustering methods can be used to create groups of pixels that are similar in regard to a measure, often their color or gray level; therefore simplifying the image by reducing the number of discrete possible pixel values. Image clustering can be used to get a simple segmentation of the image.

The Alternative Fuzzy c-Means algorithm (often abbreviated to AFcM) [9] is an iterative algorithm inspired from FcM proposed by Bezdek [8]. These algorithms find clusters in data and use the concept of fuzzy membership: instead of assigning a pixel to a single cluster, each pixel will have different membership values on each cluster.

The AFcM attempts to find clusters in the data by minimizing an objective function shown in the equation below:

$$J = \sum_{i=1}^C \sum_{j=1}^N \mu_{ij}^m d^2(x_j, c_i) \quad (1)$$

Where:

- $J$  is the objective function, a kind of quality criterion to minimize
- $N$  is the number of pixels in the image
- $C$  is the number of clusters used in the algorithm, and which must be decided before execution
- $\mu$  is the membership matrix of  $N \times C$  entries which contains the membership values of each pixel to each cluster
- $m$  is a fuzziness factor (a value larger than 1)
- $x_j$  is the  $j$ th pixel in the image
- $c_i$  is the  $i$ th cluster
- $d(x_j, c_i)$  is the distance between  $x_j$  and  $c_i$ . In the FcM, the Euclidean distance is used while in the AFcM it's defined as:

$$d^2(x, y) = 1 - e^{-\beta \|x-y\|^2} \quad (2)$$

Where  $\beta > 1$  and could be estimated from the image variance  $\bar{x}$  as:

$$\beta = \left( \sum_{j=1}^n \|x_j - \bar{x}\|^2 / n \right)^{-1} \quad (3)$$

This metric is a robustness estimator because it is insensitive to small variations and robust against noise [9].

The steps of the algorithm are:

1. Initialize  $\mu$  with random values between zero and one; but with the sum of all fuzzy membership elements for a particular pixel being equal to 1. In other words, the sum of the memberships of a pixel for all clusters must be one.
2. Estimate  $\beta$  using (3)
3. Calculate an initial value for  $J$  using (1).
4. Calculate the centroids of the clusters  $c_i$  using

$$c_i = \frac{\sum_{j=1}^n \mu_{ij}^m [1-d^2(x_j, c_i)] x_j}{\sum_{j=1}^n \mu_{ij}^m [1-d^2(x_j, c_i)]} \quad (4)$$

5. Calculate the fuzzy membership  $\mu_{ij}$  using

$$\mu_{ij} = \left( \sum_{k=1}^C \frac{[d^2(x_j, c_i)]^{1/(m-1)}}{[d^2(x_j, c_k)]^{1/(m-1)}} \right)^{-1} \quad (5)$$

6. Recalculate  $J$ .
7. Go to step 4 until a stopping condition was reached.

Some possible stopping conditions are:

- A number of iterations were executed, and we can consider that the algorithm achieved a "good enough" clustering of the data.
- The difference between the values of  $J$  in consecutive iterations is small (smaller than a user-specified parameter  $\epsilon$ ), therefore the algorithm has converged.

Traditionally the algorithm defuzzify its results by choosing a "winning" cluster, i.e. the one which is closer to the pixel in the feature space, is the one for which the membership value is highest/certain and using that cluster center as the new values for the pixel. These membership values can be obtained for any kind of images (grayscale, RGB, etc...). The algorithm is adaptive and can be used with image of multiple channels.

### B. Total Variation Regularization

The main objective is minimizing total variation for image denoising [15]. It assumes that the observed image  $f$  could be modelled as the addition of an a priori piecewise smooth (or with little variation) image  $u$  and additive noise  $v$ .

$$f = u + v \quad (6)$$

Then, the total variation of  $u$  is defined by

$$J(u) = \sum_{i,j} \| (\nabla u)_{i,j} \| \quad (7)$$

It is hence suggested to recover the original image  $u$  by minimizing  $J(u)$  by trying to solve the problem

$$\inf_u \| f - u \|^2 \quad (8)$$

where  $\| \cdot \|$  is a norm in  $L^2$ .

Chambolle [15] had demonstrated that there exists a Lagrange multiplier  $\lambda > 0$  such that, this problem has a unique solution using an algorithm to solve

$$\min_u \left[ \frac{\| u - f \|^2}{2\lambda} + J(u) \right] \quad (9)$$

This approach suits well our modelling for image compression. In fact, the coded image suffers from some data loss during the segmentation phase. This loss could be seen as error/noise added during the attribution of pixels into clusters. Thus, we can consider the model in (6) after the clustering phase and the application of TV regularization becomes a wise choice.

### III. RESULTS AND DISCUSSION

Before you begin to format your paper, first write and save the content as a separate text file. Keep your text and graphic files separate until after the text has been formatted and styled. Do not use hard tabs, and limit use of hard returns to only one return at the end of a paragraph. Do not add any kind of pagination anywhere in the paper. Do not number text heads- the template will do that for you.

In order to evaluate the effectiveness of the proposed methods, we have tested our algorithm with the help of three benchmarks of grayscale images: Lena, Baboon, and Peppers, each of size 512\*512.

There are three parameters on which our system depends on:

- The number of clusters.
- The number of iterations used in the TV algorithm.
- And  $\lambda$

It is obvious to show that using an increased number of clusters will yield better image quality but with lower compression ratio i.e. low bit-rate. In [6], we conducted a quantitative study on the compression scheme in order to achieve an acceptable bit rate with a decoded image having good quality.

In Table 1, we summarize the obtained results on our benchmarks. With Application of TV, the increase in PSNR, and hence in image quality, may reach 7% under some conditions.

In [4], the authors had shown that Lena image could be compressed using JPEG with CR = 4.39 and PSNR = 42.67 dB. Our results are very promising and the quality obtained is very close to the result obtained by JPEG. After applying TV regularization, the quality is enhanced and the PNSR increases by 4.8 % with 12 Clusters. The PSNR was 37 dB for a CR  $\approx$  4.

In [4], the authors presented a log-based technique (time consuming) that achieves a compression ratio of 14.85 on Lena image with a PSNR = 29.89. Our technique achieves a PSNR = 26 without TV and a PSNR = 27.8 after TV with a CR = 15.27. We have conducted a tiny modification on our architecture to

apply an image reduction of 75% after which we applied the compression/TV combinations. This update yields a PSNR  $\approx$  28.4 with a CR = 32.15 with 16 Clusters with immediate response time.

Table 1

	Number of Clusters	PSNR without TV (dB)	PSNR with TV (dB)	Increase in PSNR (%)	Compression Ratio	bpp
Lena	4	26	27.8	6.923	15.27	0.524
	8	32	33.7	5.313	7.31	1.094
	16	37.6	38.9	3.457	3.61	2.216
Baboon	4	25.4	27.2	7.087	5.13	1.559
	8	31	31.4	1.29	3.15	2.54
Peppers	4	25.1	26.9	7.171	13.74	0.582
	8	30.8	32.8	6.494	6.39	1.252
	16	36.8	37	0.543	3.4	2.353

PSNR is influenced by the TV parameters ( $\lambda$ , number of iterations)

In the following study, the number of classes is fixed to 8 as tradeoff between quantity and quality.

In Fig. 2 (Lena Image), we show the number of iterations affected by the value of PSNR for different values of  $\lambda$ . A maximum for the PSNR is always reached.

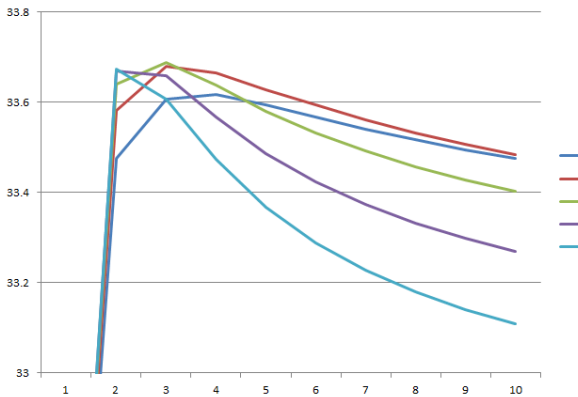


Fig. 2 Lena: The number of iterations (horizontal axis) for the TV algorithm as function of PSNR (vertical axis) for  $\lambda = 4$  to 8.

In Fig. 3 (Lena Image), we show how  $\lambda$  will affect the value of PSNR for different number of iterations in TV algorithm. A maximum for the PSNR is always reached. The algorithm requests little iteration to converge.

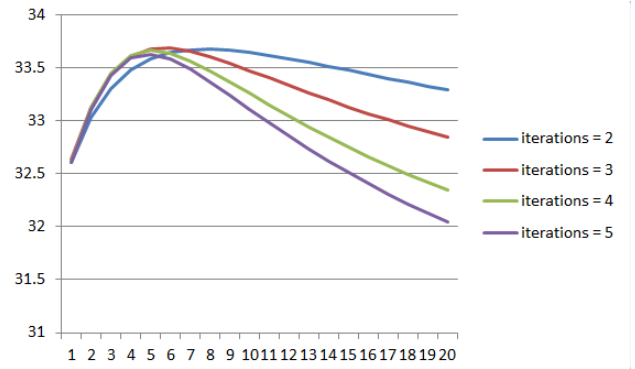


Fig. 3 the  $\lambda$  (horizontal axis) varies with the PSR (vertical axis) for iterations = 2 to 5.

In Fig. 4 and Fig. 5 (Lena Image), we show clearly that the PSNR reaches a maximum when varying  $\lambda$  and the number of iterations in both dimensions.

		Iterations									
		1	2	3	4	5	6	7	8	9	10
$\lambda$	1	31.978	32.61	32.636	32.62	32.604	32.591	32.581	32.573	32.567	32.561
	2	31.978	33.028	33.126	33.124	33.103	33.082	33.063	33.048	33.034	33.023
	3	31.978	33.301	33.438	33.452	33.433	33.409	33.386	33.365	33.347	33.331
	4	31.978	33.475	33.608	33.617	33.595	33.568	33.541	33.516	33.494	33.475
	5	31.978	33.581	33.679	33.665	33.628	33.593	33.561	33.532	33.506	33.484
	6	31.978	33.641	33.688	33.638	33.58	33.533	33.492	33.458	33.428	33.402
	7	31.978	33.669	33.658	33.567	33.486	33.423	33.373	33.332	33.298	33.269
	8	31.978	33.674	33.607	33.473	33.368	33.289	33.228	33.18	33.141	33.108
	9	31.978	33.664	33.543	33.369	33.24	33.143	33.073	33.017	32.973	32.936
	10	31.978	33.644	33.474	33.26	33.108	32.995	32.914	32.85	32.8	32.759
	11	31.978	33.617	33.403	33.151	32.979	32.849	32.757	32.685	32.629	32.583
	12	31.978	33.585	33.332	33.045	32.853	32.707	32.605	32.524	32.462	32.411
	13	31.978	33.55	33.262	32.942	32.733	32.571	32.459	32.369	32.301	32.245
	14	31.978	33.513	33.194	32.844	32.618	32.441	32.319	32.22	32.147	32.085
	15	31.978	33.475	33.129	32.75	32.508	32.317	32.186	32.078	31.999	31.932
	16	31.978	33.437	33.067	32.661	32.405	32.199	32.059	31.943	31.858	31.786
	17	31.978	33.4	33.008	32.577	32.307	32.088	31.939	31.815	31.724	31.647
	18	31.978	33.362	32.951	32.496	32.214	31.983	31.826	31.694	31.597	31.514
	19	31.978	33.326	32.897	32.42	32.126	31.883	31.718	31.579	31.477	31.388
	20	31.978	33.29	32.846	32.348	32.043	31.788	31.616	31.469	31.362	31.269

Fig. 4 PSNR is influenced by the TV parameters ( $\lambda$ , number of iterations)

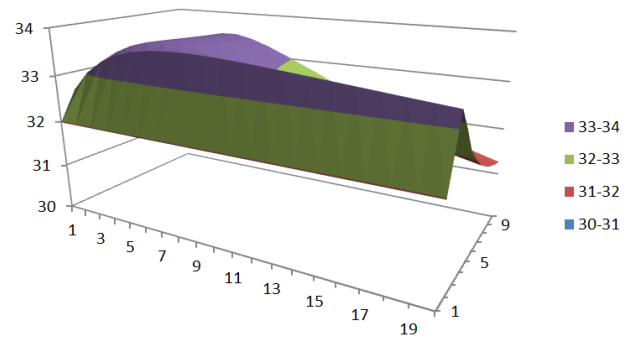


Fig. 5 3D view of the PSNR as function of parameters ( $\lambda$ , number of iterations)

#### IV. CONCLUSION

This paper presents a combination of compression/TV tools in order to offer a hybrid model for image compression/decompression schemes.

The main motivation behind our approach is to offer a decoded image with immediate access to objects/features of interest. We are targeting high quality decoded image in order to be useful on smart devices, for analysis purpose, as well as for multimedia content-based description standards.

The contribution of the study is in the application of TV regularization in the decoding phase on the segmented image in order to enhance the image quality. The experimental results had shown that TV may increase the PSNR and hence offer better quality. These results were promising and are close to the result obtained by the standard JPEG.

We have to mention that the approach depends and some parameters such as the number of clusters used during segmentation, the number of iterations and the multiplier  $\lambda$  in the TV algorithm. These parameters could be transmitted to the decoder (often in the file header) in order to set appropriately.

This approach could be enhanced in many ways. The first extension goes toward the coding of multi-channel images. We believe also that introducing ambiguity factor in the clustering phase could be beneficial for the decoding phase and especially with the use of TV regularization

#### REFERENCES

- [1] G. K. Wallace, The JPEG Still-Picture Compression Standard, Communications of ACM, Apr. 1991, pp.30-44.
- [2] N. Somasundaram and Y. V. RamanaRao, Modified LOG-EXP Based Image Compression Algorithm, IJCSNS International Journal of Computer Science and Network Security, VOL.8 No.9, Sept. 2008, pp. 179-184.
- [3] A. Gersho and R. M. Gray, Vector Quantization and Signal Compression, Kluwer Academic, 1991.
- [4] J. Weickert, Anisotropic Diffusion in Image Processing, B.G. Teubner Stuttgart, 1998.
- [5] Martin Welk, David Theis, Thomas Brox, and Joachim Weickert, PDE-Based Deconvolution with Forward-Backward Diffusivities and Diffusion Tensors, in Scale Space 2005, Springer LNCS 3459, Hofgeismar, Germany, Apr. 2005, pp. 585-597.
- [6] A.Shahin, F.Chakik, and S.Al-Ali, International Journal of Future Computer and Communication, Vol. 2, No. 3, June 2013, pp 205-209.
- [7] Lizarazo, Ivan and Barros, Fuzzy image segmentation for urban land-cover classification, Photogrammetric Engineering and Remote Sensing, Joana, 2010, pp. 151-162.
- [8] J. C. Bezdek. Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum Press, New York, 1981.
- [9] Miin-Shen Yang, Yu-Jen HU, Karen Chia-Ren-Lin, Charles Chia-Ren-Lin, Segmentation techniques for tissue differentiation in MRI of Ophthalmology using fuzzy clustering algorithms, Magnetic Resonance Imaging (20), ELSEVIER, 2002, pp.173-179.
- [10] Takahiro Saito and Takashi Komatsu, Super-resolution total-variation decoding of JPEG-compressed image data, Proc. SPIE 6502, Digital Photography III, 65020G (February 20, 2007).
- [11] Deng C W, Zhao B J. SAR image compression algorithm based on total variation decomposition, IET International Radar Conference, April.20-22, 2009, pp. 1-5.
- [12] Yuan Chen; Rong Zhang, Low Bit Rate Compression for SAR Image Based on Blocks Reordering and 3D Wavelet Transform, Sixth International Conference on Image and Graphics (ICIG), 2011, pp. 56 - 60.
- [13] Tony F. Chan and Hao-Min Zhou, Total Variation Wavelet Thresholding, Journal of Scientific Computing, August 2007, Volume 32, Issue 2, pp 315-341.
- [14] Rudin L., Osher S., Fatemi E., Nonlinear total variation based noise removal algorithms, Physica D 60, 1992, pp. 259-268
- [15] A. Chambolle, An Algorithm for Total Variation Minimization and Applications, Journal of Mathematical Imaging and Vision (20), 2004, Kluwer Academic Publishers, pp. 89-97.

**Creative Commons Attribution License 4.0  
(Attribution 4.0 International, CC BY 4.0)**

This article is published under the terms of the Creative Commons Attribution License 4.0  
[https://creativecommons.org/licenses/by/4.0/deed.en\\_US](https://creativecommons.org/licenses/by/4.0/deed.en_US)