Image Compression using POCS-based Clustering Algorithm

Le-Anh Tran, Dong-Chul Park*
Dept. of Electronics Engineering, Myongji University, Yongin, South Korea.
*Corresponding author email: parkd@mju.ac.kr

Abstract: *This paper investigates the applicability of the Projection onto Convex Set (POCS)-based clustering algorithm to image compression tasks. The POCS-based clustering approach treats all data points in a given dataset as non-intersecting convex sets and performs POCS-based parallel projections from each cluster prototype onto respective member points for the purpose of updating cluster prototypes and minimizing an objective function. The POCS-based clustering approach has been proven to be able to yield promising results against other prevailing clustering approaches in terms of convergence time and clustering error on general clustering tasks. In this study, a comparison of various clustering schemes in image compression applications has been conducted. The quantitative evaluations on various standard test images verify that the POCS-based clustering algorithm is able to perform competitively against other conventional clustering methods in image compression problems.*

Keywords: Clustering; image compression; POCS; POCS-based clustering; cluster analysis.

1. INTRODUCTION

Clustering is known as the most popular and significant task in numerous artificial intelligence-driven systems [1][2][3]. The main goal of clustering algorithms is to segregate data points from a given dataset that have similar traits and assign them to corresponding groups (or clusters). In the field of machine learning, clustering and classification problems share certain similarities, yet there is still a main difference between these two tasks. That is, classification is performed on data sets with predetermined categories (supervised learning) [4][5], whereas clustering is an unsupervised learning technique for data analysis which assigns objects into a certain number of clusters without predetermined labels [6].

General clustering approaches find homogeneous data subgroups that share identical characteristics based on some predefined criteria. One well-known objective function is the clustering error that is defined as the sum of the squared distances from cluster centers to all of their corresponding member points [7]. The K-Means method is known as one of the most classical and widely used clustering approaches for general clustering tasks that utilizes the Euclidean distance to quantify the likeness of data points [6]. The K-Means method alternates between two steps: assignment of cluster membership for all data points and recalculation of the prototypes for all clusters. The K-Means algorithm converges when the assignments of the cluster membership of all data points no longer update. However, the main weakness of the K-Means method is that its convergence process is considered heavily dependent on the data presentation sequence and the initial prototypes. On top of that, the K-Means clustering method is also considered sensitive to outliers and noisy datasets [6].

On the other hand, the Fuzzy C-Means (FCM) algorithm [8] is also known as a powerful clustering algorithm that has also been used in numerous real-world signal processing applications. Different from the K-Means method, one data point is able to belong to different clusters simultaneously in the FCM algorithm. In this sense, the confidence for a single data to be grouped into a cluster is expressed via a membership function. Similar to the K-Means method, the convergence of the FCM approach is hugely influenced by initial prototypes and initial membership parameters. In addition, expensive computational cost and incapability in processing noise and outlier are considered the main shortcomings of the FCM algorithm [8]. In order to overcome the FCM algorithm's drawbacks, the Gradient-Based Fuzzy C-Means (GBFCM) scheme has been proposed in the literature [9] which optimizes the objective function by solving two equations iteratively. The GBFCM algorithm has been proven to be competitive against the FCM algorithm for large data volume.

On the other extreme, the Projection onto Convex Set (POCS)-based clustering approach [1] is a recently introduced algorithm which has produced competitive performances against prevailing clustering methods in terms of processing time and clustering error on general clustering tasks [1][2]. With an aim to demonstrate the potential of the POCS-based approach in a wider range of applications, experiments to survey the effectuality and efficiency of the POCS-based algorithm in image compression tasks [10] are conducted in this paper. To this end, performances of various clustering methods for image compression tasks on a set of standard test image data have been compared. The empirical outcomes imply that the POCS-based clustering method is able to yield promising results against other clustering schemes in image compression problems.

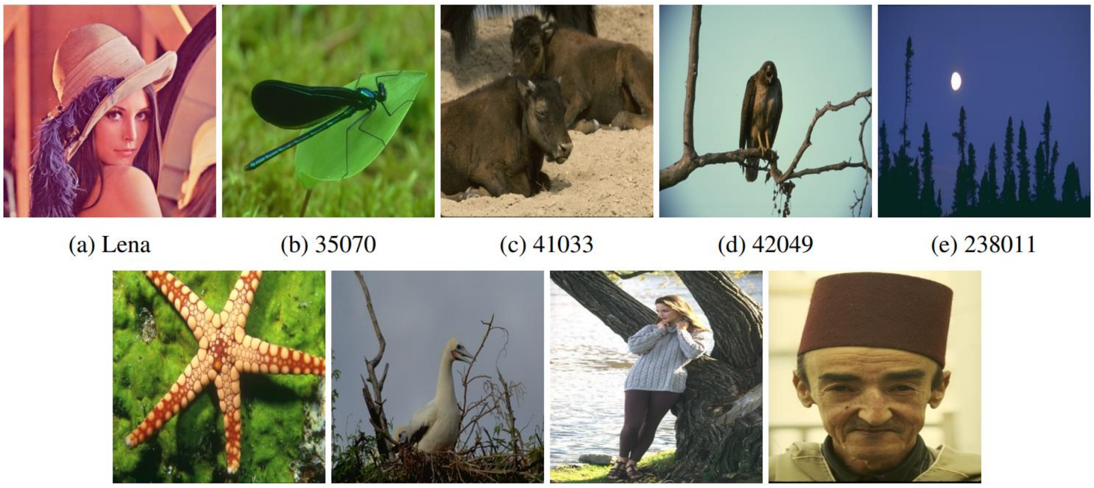
The rest of this paper is organized as follows. The POCS-based clustering approach is briefly reviewed in Section 2. The experiments and analyses are discussed in Section 3. Section 4 gives the conclusions of the paper.

2. POCS-BASED CLUSTERING ALGORITHM

This section briefly reviews the concept of convex sets, the parallel POCS method, and the POCS-based clustering approach.

2.1 Convex Set

Convex set is considered one of the most fundamental and popular concepts in the field of mathematical optimization [11]. A convex set is a set of data points that



(a) Lena (b) 35070 (c) 41033 (d) 42049 (e) 238011

(f) 12003 (g) 8049 (h) 181091 (i) 189080

Fig. 1. The images used in this study: (a) Lena image, (b-i) images from Berkeley Segmentation dataset.

satisfies the following condition: given a set A which is a subset of the Hilbert space H ($A \subseteq H$), $\forall x_1, x_2 \in A$ and $\forall \lambda \in [0, 1]$, A is called a convex set if Eq. (1) holds true:

$$x \coloneqq \lambda x_1 + (1-\lambda)x_2 \in A. \tag{1}$$

In this sense, when $\lambda = 1$, $x = x_1$, and when $\lambda = 0$, $x = x_2$. With any λ ($0 \le \lambda \le 1$), the line segment connecting two data points x_1 and x_2 is fully subsumed in A.

2.2 Projection onto Convex Set (POCS)

POCS is a simple mathematical concept that represents the projection of a point onto closed convex sets [11]. Many optimization problems have applied this method to find a point on the intersection of two or more convex sets. Given a point x (x $\notin A$), the projection of x onto a convex set A is a single point y (y $\in A$) such that y is the closest point to x. This optimization task is written as follows:

$$y = argmin\|x - y^*\|, \tag{2}$$

where y^* denotes all the data points in A.

2.3 Parallel POCS

In the parallel projection form, one data is simultaneously projected onto all the convex sets, those projections are incorporated using respective weights (importances) to solve optimization problems. For a set of n convex sets $C = \{c_i | 1 < i \le n\}$, the parallel POCS approach could be written as Eq. (3):

$$x_{p+1} = x_p + \sum_{i=1}^{n} w_i (P_{c_i} - x_p), \qquad p = 0,1,2, \dots \tag{3}$$

with a constraint:

$$\sum_{i=1}^{n} w_i = 1, \tag{4}$$

where P_{c_i} denotes the projection of x_p onto convex set c_i, w_i indicates the importance of the projection, whereas p represents the iteration index of the projection. The projection process is performed until a predefined convergence condition is met. The advantage of the parallel projection form that makes it potential in clustering task is the computational efficiency. In case of non-intersecting convex sets, the parallel POCS approach converges to a point which can minimize the weighted sum of the distances from that point to the sets, which could be written as:

$$x_{\varpi} = argmin \sum_{i=1}^{n} w_i \|x - P_{c_i}\|, \tag{5}$$

where x_{ϖ} denotes the convergence point.

2.4 POCS-based Clustering Approach

For non-intersecting convex sets, the parallel POCS approach converges to a minimum mean square error solution [2][12]. This prominent effect is applied to clustering tasks and thus the POCS-based clustering method was introduced in the literature [1]. The POCS-based clustering approach treats all data points in one dataset as non-intersecting convex sets. Given a set of data points with a predefined number of clusters k, the objective function of the POCS-based clustering method is written as:

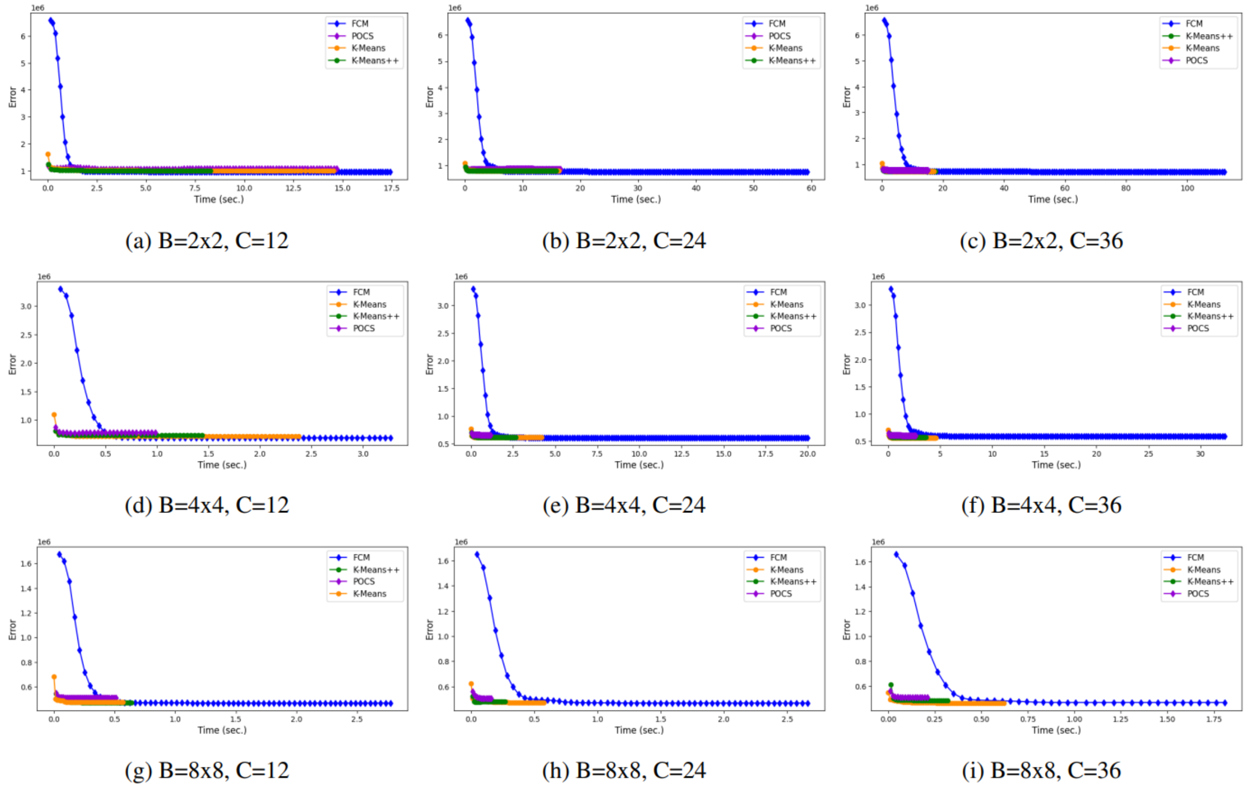
$$J = \sum_{j}^{k} \sum_{i=1}^{n_j} w_{ji} \|x_j - P_{ji}\| \tag{6}$$

in which the importance weight w_{ji} is computed as:

$$w_{ji} = \frac{\|x_j - d_{ji}\|}{\sum_{m=1}^{n_j} \|x_j - d_m\|} \tag{7}$$

where n_j represents the quantity of instances in the j^{th} cluster and P_{ji} denotes the projection from the prototype x_j onto its member data point d_{ji}.

At the beginning, the POCS-based algorithm picks k group prototypes by applying a careful seeding method [1][13]. In the subsequent step, based on the Euclidean



(a) B=2x2, C=12 (b) B=2x2, C=24 (c) B=2x2, C=36

(d) B=4x4, C=12 (e) B=4x4, C=24 (f) B=4x4, C=36

(g) B=8x8, C=12 (h) B=8x8, C=24 (i) B=8x8, C=36

Fig. 2. Typical convergence processes of various clustering methods for image compression task on Lena image.

Table 1. Comparison in terms of PSNR produced by various clustering methods with different block sizes and cluster numbers image compression task on Lena image.

	B=2x2			B=4x4			B=8x8		
	C=12	C=24	C=36	C=12	C=24	C=36	C=12	C=24	C=36
FCM	28.12	29.10	29.34	24.57	24.96	25.05	21.74	21.96	21.99
K-Means	**28.14**	**29.89**	30.84	24.79	26.07	26.83	22.27	23.42	24.01
K-Means++	27.91	29.75	30.82	24.86	25.99	26.71	22.11	23.32	24.02
POCS-based	28.05	29.88	**30.91**	**24.98**	**26.12**	**26.95**	**22.35**	**23.56**	**24.03**

Table 2. Comparison in terms of average execution time (sec.) of various clustering methods with different block sizes and cluster numbers for image compression task on Lena image.

	B=2x2			B=4x4			B=8x8		
	C=12	C=24	C=36	C=12	C=24	C=36	C=12	C=24	C=36
FCM	23.30	70.82	143.15	7.69	25.75	45.59	4.85	12.40	28.56
K-Means	**2.69**	**5.98**	**8.60**	**1.18**	2.95	3.51	0.42	0.59	0.76
K-Means++	5.52	10.02	12.39	1.31	1.73	2.52	**0.31**	0.43	0.48
POCS-based	5.35	9.92	12.42	1.30	**1.69**	**2.48**	0.32	**0.41**	**0.36**

distance, every data point is assigned to one cluster that has the prototype closest to that data point. In the last step, the POCS-based clustering method iteratively optimizes the cluster prototypes by applying Eq. (9) until there is no more update in the cluster membership of all data:

$$x_{j,p+1} = x_{j,p} + \sum_{i=1}^{n_j} w_{ji,p}(P_{ji,p} - x_{j,p})\,, p = 0,1,2,\ldots \quad (9)$$

where p indicates the iteration index. From a starting point $x_{j,0}$, the projection process converges to a point, $x_{j,\infty}$, that minimizes the objective function as in Eq. (6).

3. EXPERIMENTS AND ANALYSES

In this section, we investigate the effectiveness and efficiency of the POCS-based clustering approach in image compression tasks and compare the algorithm with other conventional cluster analysis methods such as the K-Means/K-Means++ and FCM approaches.

We first examined the performances of these clustering algorithms on only one image data but with different configurations of block size B and number of clusters C. The widely used Lena image with the resolution of 512x512 pixels, as shown in Fig. 1 (a), was utilized to conduct this experiment. Specifically, the Lena image was partitioned into blocks with different configurations

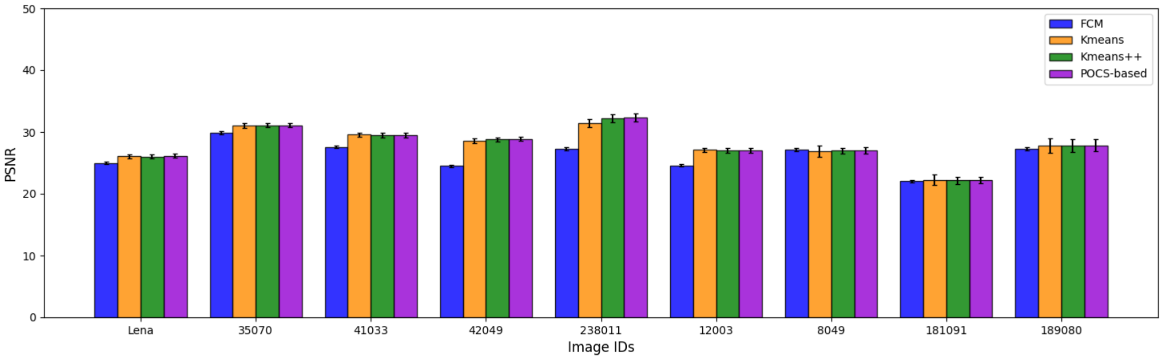


Fig. 3. Comparison in terms of PSNR produced by various clustering methods for image compression task on Lena image and the images in Berkeley Segmentation dataset (for the case of B=4x4, C=24).

Table 3. Comparison in terms of average execution time (sec.) of various clustering methods for image compression task on Lena image and the images in Berkeley Segmentation dataset (for the case of B=4x4, C=24).

	Lena	35070	41033	42049	238011	12003	8049	181091	189080
FCM	25.75	26.45	27.29	23.45	27.28	25.15	27.10	23.49	24.19
K-Means	2.95	2.56	3.53	2.37	3.36	2.55	3.55	1.37	2.54
K-Means++	1.73	**1.41**	**1.40**	1.34	1.28	1.47	1.41	1.43	0.48
POCS-based	**1.69**	1.43	1.45	**1.31**	**1.22**	**1.41**	**1.29**	**1.36**	**0.39**

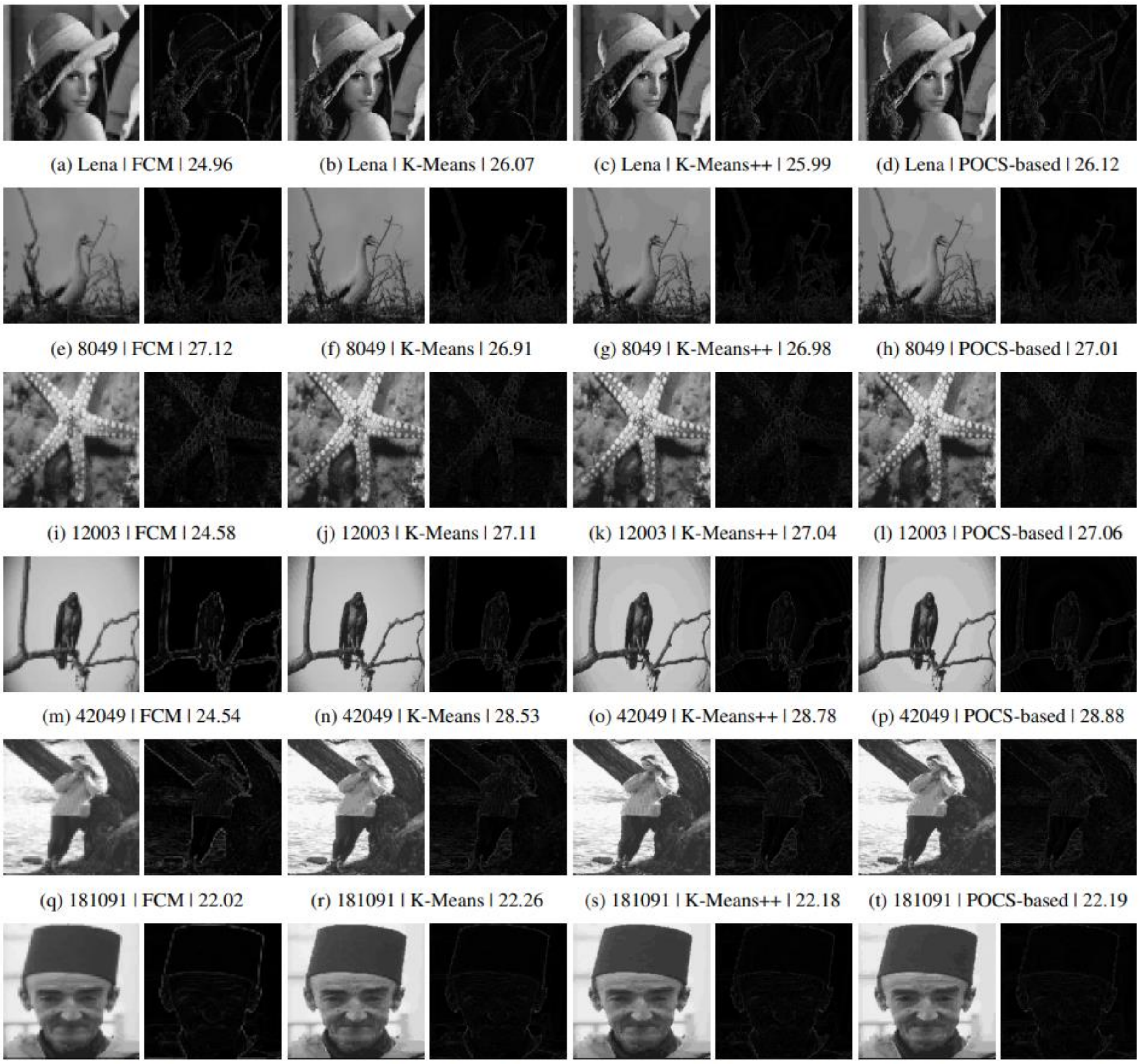
of block size, B = {2x2, 4x4, 8x8}, consequently, 65,536, 16,384, and 4,096 blocks of data were obtained for each case, respectively. We examined the performances of the aforementioned clustering algorithms in three cases of cluster numbers, C = {12, 24, 36}. Every algorithm was performed 20 times and the average measures of peak signal-to-noise ratio (PSNR) and execution time were presented and compared. The experiments were conducted on processor Intel(R) Core(TM) i5-8600K CPU @ 3.60GHz.

Table 1 and Table 2 summarize the performances in terms of mean PSNR and average execution time of various clustering methods with different block sizes and number of clusters for the image compression task on the Lena image, respectively. From the results in terms of PSNR shown in Table 1, all the algorithms can produce competitive performances compared to each other, though the PSNR results produced by the FCM algorithm are likely to degrade more drastically when the block size increases. In terms of average execution time, the FCM algorithm presents much longer execution speed compared with those of the other methods in all cases. When the block size is small (B=2x2), the K-means algorithm outperforms the POCS-based and K-means++ methods in terms of processing speed, while in the case of larger block size (B=8x8), the POCS-based clustering algorithm is able to execute faster than the K-Means approach. This stems from the difference in the prototype initializing methods of the two algorithms. Specifically, the K-means method chooses the initial prototypes at random whereas the POCS-based clustering approach adopts a careful seeding process. Moreover, in the case of B=2x2, the data amount is largest when compared to the remaining cases (65,536 data blocks), this requires more time to perform the careful seeding for initialization of prototypes. However, when B=8x8, the data volume is reduced significantly, consequently, the POCS-based approach shows better results in terms of execution speed and outperforms the K-Means approach. When compared with the K-means++ method, the POCS-based approach also produces competitive performances when B = {2x2, 4x4}. However, when the block size becomes larger (B=8x8), the POCS-based algorithm is able to adequately surpass the K-means++ method in terms of processing time. Figure 2 shows typical convergence processes of different clustering algorithms for image compression task on Lena image.

In addition to the Lena image, additional experiments on further standard test images from Berkeley Segmentation dataset [14], as shown in Fig. 1 (b-i), were also conducted. All these test images were rescaled to the resolution of 512x512 pixels. Fig. 3 and Table 3 summarize the results in terms of PSNR and average convergence time, respectively. Note that the presented results are for the case of B=4x4 and C=24 since this setting can be considered appropriate for a 512x512 input, while block sizes of 2x2 and 8x8 may be deemed as too small or too large, respectively, and 12 or 36 clusters are also known as insufficient and redundant, respectively. Similar to the previous experiments on the Lena image, the differences in terms of PSNR produced by all the examined clustering algorithms were relatively marginal. Nevertheless, the POCS-based approach demonstrates favorable performances when compared with the K-Means/K-Means++ methods and a critical advantage over the FCM approach in terms of execution speed. Overall, the outcomes of the POCS-based algorithm for the image compression tasks are satisfactory and it can be implemented into real-world image compression applications. Typical qualitative and quantitative image compression results are illustrated in Fig. 4.

4. CONCLUSIONS

This paper examines the applicability of the POCS-based clustering method to image compression problems. The POCS-based algorithm applies the convergence property



(a) Lena | FCM | 24.96 (b) Lena | K-Means | 26.07 (c) Lena | K-Means++ | 25.99 (d) Lena | POCS-based | 26.12

(e) 8049 | FCM | 27.12 (f) 8049 | K-Means | 26.91 (g) 8049 | K-Means++ | 26.98 (h) 8049 | POCS-based | 27.01

(i) 12003 | FCM | 24.58 (j) 12003 | K-Means | 27.11 (k) 12003 | K-Means++ | 27.04 (l) 12003 | POCS-based | 27.06

(m) 42049 | FCM | 24.54 (n) 42049 | K-Means | 28.53 (o) 42049 | K-Means++ | 28.78 (p) 42049 | POCS-based | 28.88

(q) 181091 | FCM | 22.02 (r) 181091 | K-Means | 22.26 (s) 181091 | K-Means++ | 22.18 (t) 181091 | POCS-based | 22.19

(u) 189080 | FCM | 27.26 (v) 189080 | K-Means | 27.79 (w) 189080 | K-Means++ | 27.80 (x) 189080 | POCS-based | 27.83

Fig. 4. Image compression results produced by different clustering algorithm: FCM, K-Means, K-Means++, and POCS-based (B = 4x4, C = 24). Each sub-figure shows the compressed image (left) and the error image (right), while the title of each sub-figure denotes {image name/ID | method | PSNR}.

of the projection onto convex sets concept to clustering and has shown competitive performances against other prevailing clustering methods. In this research, the effectiveness and efficiency of the POCS-based clustering method in real-world applications like image compression are further verified. The empirical results on a set of standard test images demonstrate that the POCS-based clustering approach is able to produce satisfactory outcomes on image compression tasks and can be considered a promising approach for image compression applications.

5. REFERENCES

[1] L. A. Tran, H. M. Deberneh, T. D. Do, T. D. Nguyen, M. H. Le, D. C. Park, POCS-based Clustering Algorithm, in Proc. of International Workshop on Intelligent Systems (IWIS), 2022, pp. 1-6.

[2] L. A. Tran, T. D. Nguyen, T. D. Do, N. C. Tran, D. Kwon, D. C. Park, Embedding Clustering via Autoencoder and Projection onto Convex Set, in Proc. of International Conference on System Science and Engineering (ICSSE), 2023, pp. 62-66.

[3] L. A. Tran, M. H. Le, Robust U-Net-based Road Lane Markings Detection for Autonomous Driving, in Proc. of International Conference on System Science and Engineering (ICSSE), 2019, pp. 62-66.

[4] M. Jang, D. C. Park, Classifier Integration Model with Confusion Table, in Proc. of International Exchange and Innovation Conference on Engineering & Sciences (IEICES), 2018, pp. 144-146.

[5] H. A. Munira, M. S. Islam, Multi-Classification of Brain MRI Tumor Using ConVGXNet, ConResXNet, and ConIncXNet, in Proc. of International Exchange and Innovation Conference on Engineering & Sciences (IEICES), 2022, pp. 169-175.

[6] J. MacQueen, Some methods for classification and analysis of multivariate observations, in Proc. of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1 (Univ. of Calif. Press, 1967), pp. 281-297.

[7] A. Likas, N. Vlassis, J. J. Verbeek, The global k-means clustering algorithm, Pattern Recognition, Volume 36, 2003, pp. 451-461.
[8] J. C. Bezdek, R. Ehrlich, W. Full, FCM: The fuzzy c-means clustering algorithm, Computers and Geosciences, Volume 10, 1984, pp. 191-203.
[9] D. C. Park, I. Dagher, Gradient based fuzzy c-means (GBFCM) algorithm, in Proc. of 1994 IEEE International Conference on Neural Networks (ICNN'94), Volume 3, IEEE, 1994, pp. 1626–1631.
[10] D. C. Park, Y. J. Woo, Weighted centroid neural network for edge preserving image compression, IEEE Transactions on Neural Networks, Volume 12, 2001, pp. 1134-1146.
[11] L. Bregman, The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming, USSR Computational Mathematics and Mathematical Physics, Volume 7, Issue 3, 1967, pp. 200–217.
[12] R. Y. Albert, R. J. Marks II, K. E. Schubert, C. Baylis, A. Egbert, A. Goad, S. Haug, Dilated POCS: Minimax Convex Optimization, IEEE Access, Volume 11, 2023, pp. 32733-32742.
[13] D. Arthur, S. Vassilvitskii, K-means++: the advantages of careful seeding, in Proc. of The Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, Jan. 2007, pp. 1027–1035.
[14] P. Arbelaez, M. Maire, C. Fowlkes, J. Malik, Contour Detection and Hierarchical Image Segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 33, 2011, pp. 898-916.