that their inverse function leads to multiple solutions during occurrence of transformation. This may require selections or development of (inverse) criterion for taking decision on selecting right solution values to continue with inverse algorithms (for retrieving original image information). It will definitely increase the volume of data to be interpreted. It is not covered in this paper as the formulation is yet to be satisfactorily tested.

Returning to issue of interest, the argument for image partitioning of an image is the certain basic features within image that are radically present in regions of image. These distinct features can be appropriately used for discretizing the image on larger scale thus leading to partitioning.

Thus, it may well assumed certain feature is distinctively present in one partition of image; while another partition might be involving a feature different from the previous one. Such characterizing features may be represented using Lipchitz (functions, properties).

- Let f_1, f_2, f_3 etc be Lipschitz such that $f_1, f_2, f_3: X \to X$, then f_1 be characterizing partition-1 of image while f_2 be characterization partition-2 of image and so on.
- The proper identification of Lipschitz function in image space can lead to appropriate partitioning strategy for given image.
- Occurrence of more than one function (f_1, f_2 etc) in image space signifies some level of discontinuity from the perspective of individual functions i.e. partition space characterized by function f_1 (say) might have discontinuous f_1 at partitioning line or curve.
- This simply means characteristic function in one partition space may experience discontinuity in mathematical sense or significant jump in value towards/at partition curve.
- This viewpoint may be extended to various functions (f_1, f_2 etc) in different partition space.
- Individual partitions can be treated as closed and bounded from metric space point of view with regard to non-empty set X. Let $A \subset \mathbb{R}^2$ also belongs to a partition in image space.
- Thus there also occurs ε-neighbourhood of A represented by $S_{A,\varepsilon}$ (Fig. 5). If A is closed and bounded subset of $\mathbb{R}^2$ then $S_{A,\varepsilon}$ is also closed and bounded.
- Now Hausdorff space $H(X)$ can be defined considering individual partitions of image.

Let $A, B \in H(X)$, then Hausdorff metric can be appropriately defined as

$$\hbar(A,B) = max\left\{inf\{\varepsilon: B \in S_{A,\varepsilon}\}, inf\{\varepsilon: A \in S_{B,\varepsilon}\}\right\}$$

where $S_{A,\varepsilon}$ is the ε-neighbourhood of A and $S_{B,\varepsilon}$ is the ε-neighbourhood of B

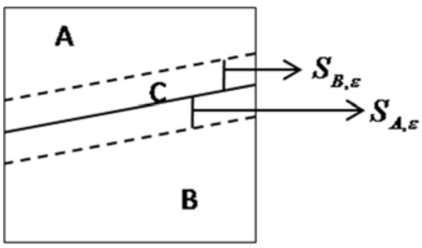

Fig. 5: ε-neighbourhood

- Thus $S_{A,\varepsilon}, S_{B,\varepsilon}$ holds not only the key for identification of partition curve but also for other characterizations of partition space. This provides better flexibility for simultaneous considerations of more than one Lipschitz (i.e. f_1, f_2, f_3etc.) and possibly a way to connect or collectively interpret more partition space. (There seems little reason, at least at this stage, to doubt as why not all partitions in an image handled simultaneously).

Here we appropriately treat image properties like multiple colors, contrasts, brightness etc. Each property is identified through functional information or distribution. Property related information can be adjusted and checked for partitioning possibilities. Related descriptions favor functions that are continuous on $\mathbb{R}^2$. Often a region A of $\mathbb{R}^2$ is considered to be closed and bounded subset of $\mathbb{R}^2$. We can also consider many subsets of $\mathbb{R}^2$ each of which are closed and bounded. Let their union,$\cup_{i=1}^{n} A_i$ be completely describing the input image (region). Indicated union operation ascertains flexibility that $A_i's$ may be overlapping each other but in the proposed algorithm the overlapping aspect of the clusters is not considered (but left for future work).

4. Application

Methodology is illustrated through algorithm given below.

The following discrete steps are involved:

1. Input a grey image I
2. Convert I to pixel values $p_{i,j} \in [0,255]$, zero being black, 255 being white in between there are shades of grey.
3. The pixel value of each cell is compared with at most n^2 neighboring cells (n each horizontally and vertically) and if the value matches to tolerance t, a cluster 1 is engraved out with partitioning curve C_1
4. The process is repeated unless the last cell is reached
5. If any cluster $\mathcal{C}_r \subset \mathcal{C}_s$, then reject $\mathcal{C}_r$
6. Find the centre of distribution (L) of all clusters individually.
7. Replace $p_{i,j}$ with $L \,\forall\, i,j \in \mathcal{C}_r$.
8. With new pixel values construct the image.
9. Output image I'.

While forming the clusters, the outliners (if any) in the adjacent positions are not included in any of the clusters and will be left unprocessed. The composition of mappings are used and the centre of distribution is defined by the function $f: \mathbb{I} \to \mathbb{I},\ \in [0,255]$, given as $f(A_i) = \mathscr{f}(\mathscr{g}(A_i)) = B_i \quad \forall\, A_i \in \mathcal{C}$, where $A_i = \cup\, p_{i,j}$ is the set of all pixels in a cluster $\mathcal{C}_i$, $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2 \ldots \ldots \mathcal{C}_q\}$, q=number of clusters, $\mathscr{f}$ = floor function

$$\mathscr{g}(A_i) = \frac{1}{N}\sum_1^N x_i \quad \forall\, x_i \in A_i$$

f is a contraction on the non-empty set of pixel points $p_{i,j} \in [0,255]$. The distance between two clusters is calculated in Root Mean Square sense, metric ρ is taken to be the quality metric PSNR.
Each pixel in a cluster (region) is similar with respect to some characteristic or computed property, such as color, texture or intensity. Human vision is receptive to edges hence keeping the factor n small will preserve the informative edges. Above algorithm is implemented on various 64×64 test images (Fig. 6a, b, c). Steps 1, 2, 8 and 9 of the algorithm are implemented using python on spyder IDE, remaining of the computation is done on MS excel. The algorithm can be tweaked or modified to suit typical software like python language and digital images with parametric variety can be processed.

5. Result and analysis

Fig. 6a, b, c displays the test image I (original) and 7a, b, c and 8a, b, c are the corresponding output images I' (compressed or data reduced). The result obtained for the test images taking $n = 8$, $t = 30$ then taking $n = 8$, $t = 10$ is summarized in the table 1.

Fig.6 Test image a b c

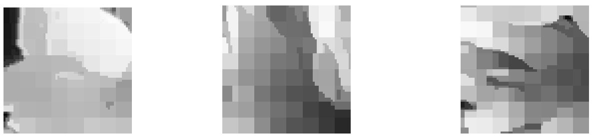

Fig.7 Output image a b c

Fig.8 Output image a b c

	Figure	Space (KB)	No. of clusters	PSNR
Test Image	6a	12.4	--	--
	6b	12.7	--	--
	6c	13	--	--
Output Images $n = 8, t = 30$	7a	0.73	134	33.37
	7b	0.54	95	25.33
	7c	0.68	125	23.48
Output Images $n = 8, t = 10$	8a	1.50	343	39.71
	8b	1.94	528	39.66
	8c	2.22	586	39.34

Table1: Test summary

The method is lossy which means the output image is approximate to the original image that is the matrix representing I' is close to matrix of I. Quality of the output images when compared to the corresponding originals is obtained as PSNR. Preliminary inquiry indicates that increasing the number of clusters by decreasing n and/or tapering the tolerance (t) is likely to improve the PSNR but will not be space efficient. For optimal partitioning curve, optimum number of clusters is required both in terms of storage space and time consumption during search process for clustering and image reconstruction.

6. Conclusion

An algorithm for partitioning the digital images into clusters of variable sizes that takes into account factors n (dimension of square block of pixels) and t (tolerance in terms of the range of pixel values within the square block) has been developed based on Contraction mapping. The centre of distribution of the cluster is defined such that it facilitates partitioning and assists in compression process. The results indicate implementation success for the proposed algorithm. Comparison of input and the output images demonstrate reduction in image data by preserving the important content and structures of the original image as anticipated. The method is lossy, though. Novel algorithm has potential to downsize the larger sized images to suit data streaming for smaller display gadgets like phones, tablets etc.

References

[1] Banach, S. (1922): Sur les Op´erations Dans les Ensembles Abstraits et Leur application aux ´equations int´egrales, Fundamenta Mathematicae, **(3)**, pp. 133-181.
[2] Barnsley, M. F. (1988): Fractals everywhere, Academic Press, New York.
[3] Barnsley, M. F.; Sian, A. A. (1988): Better way to compress images, BYTE, pp. 215- 223.
[4] Bosilj, P.; Kijak, E.; Lefèvre S.(2018): Partition and Inclusion Hierarchies of Images: A comprehensive survey, J. of Imaging, **4 (2)**, pp.1-31.
[5] Boxer, L. (1994): Digitally continuous functions, Pattern Recognition Letters, **15**, pp. 833-839.
[6] Boxer, L. (1999): A classical construction for the digital fundamental group, J. Math. Imaging Vis., **10**, pp. 51-62.
[7] Diane, W. N. N; Xingming, S.; KueMoise, F. (2014): A survey of partition-based techniques for copy-move forgery detection, The Scientific World J., Article ID 975456, 13 pages.
[8] Ege, O.; Karaca, I.(2015): Banach fixed point theorem for digital images, J. Nonlinear. Sci. Appl., **8**, pp. 237-245.
[9] Falconer, K.(1952): Fractal geometry: Mathematical foundations and applications, John Wiley, New York.
[10] Fisher, Y. (1994): Fractal image compression-Theory and applications", Springer Verlag, New York.
[11] Hutchinson, J. E. (1981): Fractals and self-similarity, Indiana Univ. Mathematics J., **35 (5)**, pp. 713-747.
[12] Jacquin, A. E. (1992): Image coding based on a fractal theory of iterated contractive image transformations, IEEE Transactions on Image Processing, **1 (1)**, pp. 18-30.
[13] Jayasankar, U.; Thirumal, V.; Ponnurangam D. (2021): A survey on data compression techniques: From the perspective of data quality, coding schemes, data type and applications, J. of King Saud Univ. – Computer and Information Sciences, **33**, pp. 119–140.
[14] Reusens, E. (1994): Overlapped Adaptive partitioning for image coding based on the theory of iterated function systems, IEEE proceedings of ICASSP, **5**, pp. 569-572.
[15] Yuan, X.; Cai, Z. (2019) "An adaptive triangular partition algorithm for digital images, IEEE Transactions on Multimedia, **21 (6)**, pp. 1372-1383.