Tabular Approach in *K-means* Clustering

Victor J. D. Tsai

Department of Civil Engineering, National Chung Hsing University, Taiwan, jdtsai@nchu.edu.tw

ABSTRACT ... The functionalities of practical image data handling systems are always computational intensive and are limited by the computing performance of the hardware devices, limitation and deficits of the software, and deficiency in processing large volume of remote sensing images. This research aims on developing a Tabular *K-means* clustering using Visual C++. The basic idea of traditional *K-means* approach was refined by deriving a look-up table (LUT) from the Voronoi diagram of the automatically detected peaks in the scatter diagram of the first two principal components of the images. The performance of numerical experiments in clustering 7-band Landsat thematic mapper (TM) images into specified number of spectral clusters is demonstrated for the advantages in computational efficiency of the proposed approach against traditional approach in *K-means* clustering.

KEY WORDS: Clustering, *K-Means*, Scatter Diagram, Peak Detection, Voronoi Diagram, Look-Up Table

1. INTRODUCTION

In remote sensing image classification, clustering is the method of grouping pixels into spectral sets so that pixels within the set should have high similarity. Generally, the functionalities of practical image data handling systems are always computational intensive and are limited by the computing performance of the Central Processing Unit (CPU), memory allocation, capacity of storage device, limitation and deficits of the software, and deficiency in process in processing large volume of remote sensing images. Since there is no theoretical limitation on the number of bands (or features), in principle, any of the conventional algorithms for classifying multispectral data can be directly applied to hyperspectral data. However, these algorithms appear suddenly inefficient when applied to a hyperspectral image with 200 or more closely related spectral bands (Schowenderdt, 1997). The increased number of spectral bands results in a vast increase in the computational load for statistical analysis in order to derive reliable class-specific statistics for maximum-likelihood approach. The access to efficient hardware and software is an important factor in determining the ease with which an unsupervised classification can be performed (Lillesand *et al.*, 2015).

Image data clustering in remote sensing is usually referred as unsupervised classification based on the natural groupings in the image values of spectral reflectance. Clustering algorithms can be broadly categorized into two groups: *hierarchical* and *partitional* (Jain, 2010). Among the *partitional* algorithms, *K-means* is the most popular one due to its simplicity, efficiency, and empirical success in recognizing multivariate data. There have been many variants of *K-means* clustering since it was discovered in the 1950's. This research aims on developing a Tabular *K-means* approach using Visual C++. The basic idea in traditional *K-means* approach (Duda & Hart, 1973) was examined and refined with principal component transformation (PCT), peak detection and Voronoi diagram in clustering remote sensing images into specified number of spectral clusters. Experiment results from clustering 7-band Landsat thematic mapper (TM) images demonstrated the advantages in computational efficiency of the proposed Tabular *K-means* approach against traditional method.

2. PRINCIPLES IN K-MEANS CLUSTERING

2.1 Discriminant Functions

Among the algorithms of unsupervised classification, the notion of similarity between a pixel and clusters is a fundamental concept behind these classifiers. The measure of similarity between a pixel $\boldsymbol{x}$ to a cluster ω_j with known class signature in a classifier can be defined by choosing one of the following commonly-used distance measures (Richards & Jia, 2006; Gonzalez & Woods, 2008; Schowengerdt, 1997; Jenson, 2005):

A. Manhattan (City-block) distance:

$$D_C(x, m_j) = \left| x - m_j \right| = \sum_{l=1}^{L} \left| x_l - m_{jl} \right| \quad (1),$$

B. Euclidean distance, represented in squared norm of the difference vector for being monotonic:

$$D_E(x, m_j) = \left\| x - m_j \right\|^2 = \left(x - m_j \right)^T \left(x - m_j \right) \quad (2),$$

C. Mahalanobis distance, which is a multivariate generalization of the Euclidean distance for Gaussian normal distribution:

$$D_M(x, m_j) = \left(x - m_j \right)^T C_j^{-1} \left(x - m_j \right) \quad (3),$$

where, for an *L*-band image,

$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_L \end{bmatrix}$ represents the pixel in consideration, and

$$m_j = \begin{bmatrix} m_{j1} \\ m_{j2} \\ \vdots \\ m_{jL} \end{bmatrix} = \frac{1}{N_j} \sum_{i=1}^{N_j} x_i \quad \text{for } x_i \in \omega_j \tag{4}$$

$$C_j = \frac{1}{N_j - 1} \sum_{i=1}^{N_j} (x_i - m_j)(x_i - m_j)^T \quad \text{for } x_i \in \omega_j \tag{5}$$

represent the mean vector and the covariance matrix of a cluster ω_j with N_j pixels, respectively.

Clearly, the Manhattan distance is computationally the fast with the least accuracy among the three distance measures. Though the Mahalanobis distance is the slowest in computation, it is the most accurate in considering the practical nature of pixel distribution within a cluster. For K clusters, the basic problem in clustering analysis applying *minimum distance classifier* is then to find K discriminant functions of selected distance measure with the property that

$$x \in \omega_j \text{ if } D(x, m_j) < D(x, m_i) \;\; i, j = 1,2,...,K; i \neq j \tag{6}.$$

Note that the well-known *maximum likelihood classifier* can be degeneralized to the *minimum distance classifier* that employs Mahalanobis distance measure on the basis of equal prior probabilities (Richards & Jia, 2006).

2.2 The *K-means* Algorithm

K-means algorithm (Duda & Hart, 1973) is one of the most commonly clustering methods (Lillesand *et al*., 2015). It comprises the following iterative process for clustering multivariate data into groups with similar properties:

A. Arbitrarily assign initial mean vector ("seed" or "attractor") for each of the K clusters,
B. Arbitrarily assign initial covariance matrix for each of the K clusters if D_M is employed,
C. For each pixel in the scene,
   C.1 compute the K discriminant functions based on selected distance measure, i.e., Eq. (1), (2), or (3),
   C.2 assign this pixel to a cluster according to Eq. (6),
D. Update the mean vectors of all K clusters according to Eq. (4), and update the covariance matrices of all K clusters according to Eq. (5) if D_M is employed,
E. Reset the number of pixels for each cluster to zero, and repeat steps C to D until there is no significant change in pixel assignments,
F. Output the result of clustering with the cluster assignment to all pixels and the spectral signatures, i.e., mean vectors and covariance matrices, of all clusters.

The common criterion for ending the iterative process in step F can be defined in terms of (1) the *net mean migration* (Δm) (Schowengerdt, 1997), i.e., the magnitude change of the mean vectors over all K clusters, from iteration i to the previous *i-1*, or (2) the *sum of squared error* (*SSE*) (Richards & Jia, 2006) for all the clusters as the followings:

$$\Delta m(i) = \sum_{j=1}^{K} \left\| m_j^i - m_j^{i-1} \right\| = \sum_{j=1}^{K} \left(D_E(m_j^i, m_j^{i-1}) \right)^{1/2} \tag{7},$$

$$SSE(i) = \sum_{j=1}^{K} \sum_{x \in \omega_j} D_E(x, m_j) \tag{8}.$$

As $\Delta m(i)$ reaches zero or $SSE(i)$ converges to $SSE(i\text{-}1)$, the iterative procedure is terminated. A threshold of the criterion or a maximum number of iteration may be set to avoid timely iterative process.

3. THE TABULAR K-MEANS APPROACH

This paper employs techniques in principal component transformation (PCT), peak detection on scatter diagram, and Voronoi diagram of the selected peaks in preparing a look-up table (LUT) for *K-means* clustering of remote sensing multispectral and hyperspectral images. PCT is a feature space transformation designed to remove the high spectral redundancy in multispectral and hyperspectral image bands (Tsai & Tsai, 2013). PCT was applied to use only the first two principle components (PCs) that contain the most variance for spectral clustering. The two-dimensional (2-D) scatter diagram of the first two PCs, considering the two-channel pixel values as positional coordinates, was computed for selecting the K initial seeds by applying 2-D peak detection techniques. Given the set of K seeds, the raster-based Voronoi diagram is employed to partition the 2-D discrete grids of the scatter diagram into K convex Voronoi cells, i.e., clusters, that are closer to one seed than to any other seeds. A look-up table (LUT) was then generated for the 2-D discrete grid for fast mapping of each pixel into a cluster assignment.

Therefore, steps A to C of the traditional *K-means* algorithm were modified in the proposed Tabular *K-means* approach as following:

A. Assign initial mean vector for each of the K clusters:
   A.1 Transform the hyperspectral image into principal components (PCs) with its eigenvalues in descending order (Tsai & Tsai, 2013),
   A.2 Compute the 2-D scatter diagram of the first two PCs,
   A.3 Find the highest K peaks, which are apart from each other by s specified distance threshold, from the 2-D scatter diagram as initial mean vectors.
B. Compute initial covariance matrix for each of the K clusters if D_M is employed,
C. For the first two PCs,
   C.1 Compute the 2-D Voronoi diagram of the K mean vectors in the 2-D spectral space,
   C.2 Assign each pixel to a cluster by the look-up table using its digital values of the two PCs as indices.

In this case, the initial *K* seeds are much closer to the centroids of actual groups of pixels with similar spectral characteristics than those arbitrarily assigned. As a result, the iterative clustering may converge fast in a limited number of iteration.

4. EXPERIMENTAL RESULTS

The programs were developed using Microsoft Visual Studio 2012 C++ under Microsoft Windows 7 Enterprise 64-bit environment in an ASUS U36J series Notebook with Intel Core™ i5-M460 CPU @2.53GHz and 4GB RAM. The C++ codes were developed as Win32 Console applications for both traditional *K-means* approach and the proposed Tabular *K-means* approach for comparison of the computational performance in terms of convergent and run-time efficiency.

As shown in Fig. 1(a), a 7-band, *256*×256, 8-bit Landsat TM image of Taichung area in Taiwan was used in the experiment. The Landsat TM image was transformed into PCs with the first two PCs (PC-1 and PC-2) shown in Fig. 1(b) and 1(c). The results of PCT on the experiment images show that PC-1 and PC-2 together contain 95.96% (66.45% + 29.51%) of the spectral variance of the original 7-band images. Thus, only PC-1 and PC-2 were used in computing the scatter diagram from which the desired number of separate peaks of local maximum were automatically detected and whose coordinates (pixel values) were used as the initial seeds. A *256*×256 raster Voronoi diagram of these seeds, for each iteration, was then generated for use as the LUT that actually claims the grouping region of each seed in the spectral intensity space. Fig. 2 illustrate the transition of the Voronoi diagram through iterations in the proposed approach with 20 initial seeds, whose means and standard deviations were updated accordingly. Fig. 3 shows the cluster map of both approaches with 20 seeds.

The performances of the proposed Tabular *K-means* and the traditional *K-means* are shown in Tables 1. It demonstrates that (1) the SSE decreases and the runtime per iteration grows as the number of seeds grows for both approaches, and (2) the proposed Tabular *K-mean* approach dominate traditional *K-mean* approach in efficiency in terms of runtime, both total and per iteration.

5. CONCLUSIONS

This research adapts Visual C++ in designing programs for unsupervised *K-means* clustering analysis, and compares the computational efficiency. We focused on the developing a Tabular *K-means* algorithm with minimum distance classifier. A 7-band Landsat TM image were used in the experiments for illustrating that the proposed Tabular *K-means* approach dominates the traditional approach through the use of LUT from employing Voronoi diagram of the seeds. It is anticipated that parallelism of basic operations in the proposed Tabular *K-means* on multi-core multi-thread CPUs and GPUs for high performance processing of volumetric image data in remote sensing applications.

ACKNOWLEDGEMENTS

This research was supported by the Ministry of Science and Technology, Executive Yuan, Taiwan, under the contract grant NSC 100-2221-E-005-075-MY2.

REFERENCES

Duda, R. D. and O. E. Hart, 1973. *Pattern Classification and Scene Analysis*. John Wiley & Sons.

Gonzalez, R. C. and R. E. Woods, 2008. *Digital Image Processing*, 3rd ed., Pearson Prentice Hall.

Jain, A. K., 2010. Data clustering: 50 years beyond K-means, *Pattern Recognition Letters*, 32, pp. 651-666.

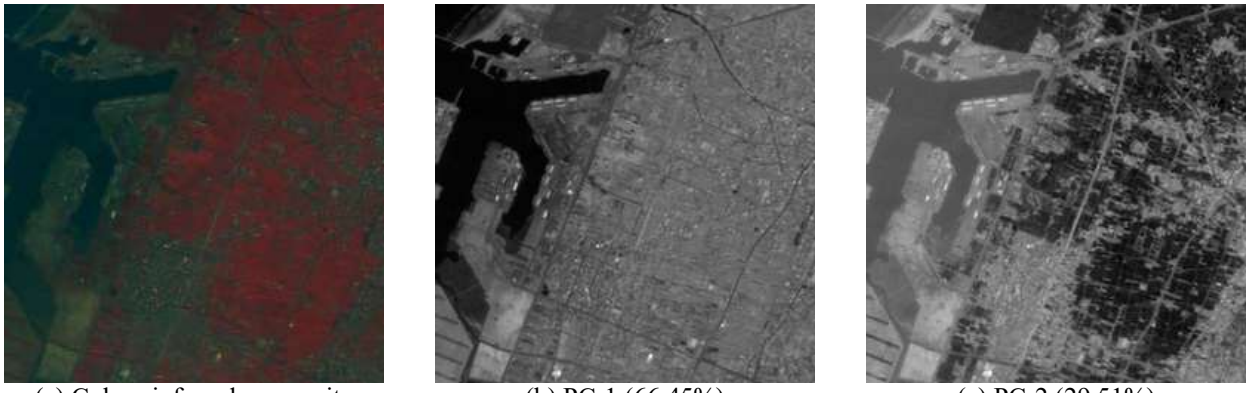
Jenson, J. R., 2005. *Introductory Digital image Processing: A Remote Sensing Perspective*, 3rd ed., Pearson Prentice Hall.

Lillesand, T. M., R. W. Kiefer, and J. W. Chipman, 2000. *Remote Sensing and Image Interpretation*, 7th ed., Wiley.

Richards, J. A. and X. Jia, 2006. *Remote Sensing Digital Image Processing: An Introduction*, 4th ed., Springer.

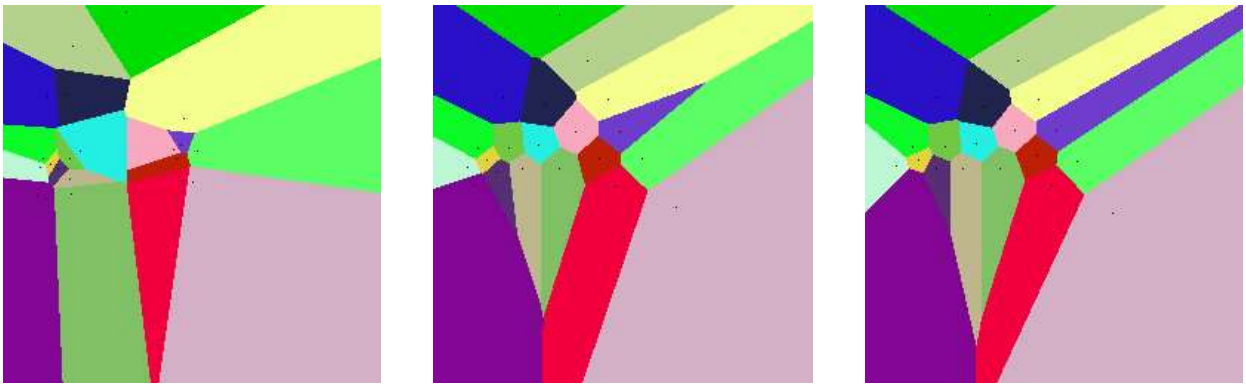
Schowengerdt, R. A., 1997. *Remote sensing: Models and methods for image processing*, Academic Press.

Tsai, V. J. D. and C. W. Tsai, 2013. GPU-Based Parallelization on Principal Component Transformation, *ISRS 2013*, Chiba, Japan, 4 p.

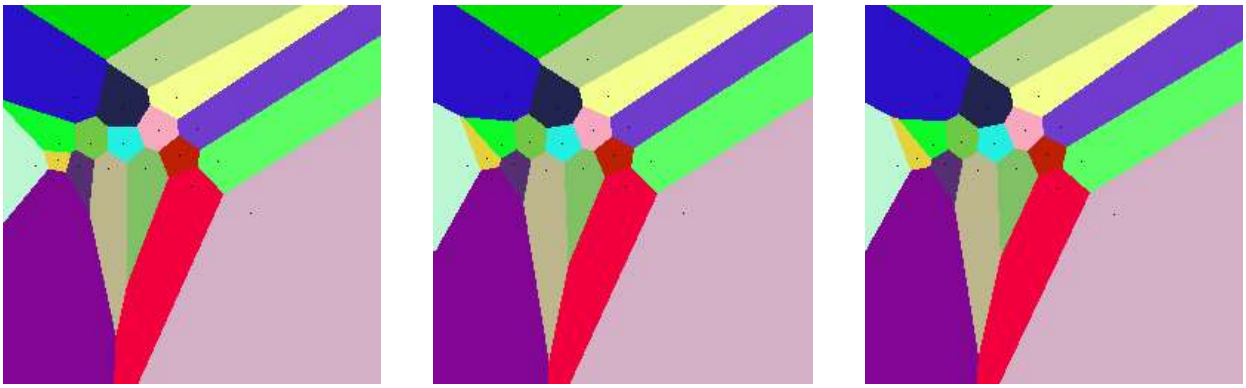


(a) Colour infrared composite (b) PC-1 (66.45%) (c) PC-2 (29.51%)

Fig. 1. Landsat TM images (8-bit, *256*×256 in size) and its principal components used in the experiment.



(a) iteration 1 (initial) (b) iteration 7 (c) iteration 14



(d) iteration 20 (e) iteration 27 (f) iteration 33 (final)

Fig. 2. Transition of the Voronoi diagrams of 20 seeds (black dots) for the Landsat TM images.

(a) Tabular *K-means* (b) Traditional *K-means*

Fig. 3. Cluster maps of 20 seeds from the Landsat TM images.

Table 1. Performance of the proposed Tabular K-means and the traditional *K-means*

Method	**Tabular *K-means***				Traditional *K-means*			
K	# iteration	SSE	runtime (msec)	runtime / iteration	# iteration	SSE	runtime (msec)	runtime / iteration
5	25	25184350.16	314.197	12.568	50	17979600.92	2016.660	40.333
10	53	11404443.93	741.676	13.994	55	9719515.63	3556.816	64.669
15	60	7629334.88	1027.845	17.131	111	7766599.72	9866.895	88.891
20	33	5776961.06	706.096	21.397	124	6402920.79	14041.155	113.235
25	34	4894041.88	837.767	24.640	425	5575943.36	57752.073	135.887
30	22	4208016.73	649.435	29.520	175	5075002.08	27961.862	159.782
35	20	3708905.17	660.882	33.044	198	4674793.70	36143.634	182.544
40	31	3248283.50	1074.310	34.655	134	4376743.73	27647.488	206.324
45	51	2919433.75	1865.670	36.582	212	4115886.28	49059.936	231.415
50	30	2651855.30	1223.050	40.768	183	3990616.01	46473.340	253.953