High-Resolution Satellite Imagery Changes Detection using Agglomerative Fuzzy *K*-Means Clustering Algorithm

C. Pandimuthu
System Programmer
Dept. Of Computer Science and Engineering
Alagappa University
Karaikudi, India

K. Kuppusamy
Professor
Dept. Of Computer Science and Engineering
Alagappa University
Karaikudi, India

ABSTRACT

The high-resolution commercial satellite imagery (HRCSI) has increased significantly over the last 5 years for a wide variety of applications. This has increase in volume, frequency of acquisition, and spatial resolution of HRCSI. In particular, satellite images contain land cover types; large areas (e.g., building, bridge and roads) occupy relatively small regions. The change detection and exploitation of change between multi temporal high-resolution satellite and air bone images. Overlapping multi temporal images are first organized in to 256m x 256m tiles in a global grid reference system. The tiles are initially ranged by these changes scores for retrieval, review, and exploitation in web based applications. Automatically detecting regions or clusters of such widely varying sizes is a challenging task. In this paper we present an agglomerative fuzzy *K*-Means clustering algorithm in change detection. The algorithm can produce more consistent clustering result from different sets of initial clusters centres, the algorithm determine the number of clusters in the data sets, which is a well – known problem in *K*-means clustering.

Index Terms- High-Resolution satellite imagery, Change detection, clustering, agglomerative, Fuzzy K-means clustering cluster validation.

1. Introduction

The High-Resolution Satellite Imagery (HRSI) has grown tremendously in the last few years. The commercial markets for online mapping, personal navigation, etc., one important application for multi temporal HRCSI is change detection. Information derived from changes in land cover is used in application including emergency response and management, environmental monitoring, urban growth assessment and planning, and defence and intelligence surveillance. The multi temporal high-resolution satellite and air bone images called Geo CDX. It is stands for Geographical Change Detection and Exploitation [1], the images divides into 256m x 256m tiles in a worldwide UTM-based grid reference system a tile change score calculated as an aggregation of individual pixel-level change scores. Clustering is an unsupervised method for partitioning a data set into groups containing similar data. Clustering algorithms have been previously utilized in various types of geospatial image processing. For example, utilization of clustering algorithms is to discover different classes of land cover in a geospatial image. Tyagi *et al.* [2] proposed a context-sensitive clustering approach using graph-cut initialization and an *expectation maximization* (EM) algorithm for classifying pixels from a multispectral (MS) Landsat-5 image into different classes of land cover, while Maulik and Saha [3] proposed the use of modified differential-evolution-based fuzzy clustering. Yang *et al.* used the *fuzzy statistics similarity* as a metric in an *affinity propagation* clustering algorithm to extract land cover information from Landsat-7, Quick bird, and MODIS data sets [4]. Another utilization of clustering is in change detection between two multi temporal geospatial images. Celik [5] employed *c-means* clustering and *principal component analysis* to perform change detection on multi temporal satellite imagery. Gosh *et al.* [6] found that change detection of multi temporal satellite imagery using *fuzzy c-means* (FCM) and *Gustafson–Kessel* clustering algorithms produced better results than those obtained using *Markov random field* and other neural-network-based algorithms. Clustering is a process of grouping a set of objects into clusters so that the objects in the same cluster have high similarity but are very dissimilar with objects in other clusters. Various types of clustering methods have been proposed and developed; see, for instance, [1]. K-Means algorithm have been reported by Ruspini [4] and Bezdek [6], where each pattern is allowed to have memberships in all clusters rather than having a distinct membership to one single cluster. Numerous problems in real world applications, such as pattern recognition and computer vision, can be tackled effectively by the fuzzy K-Means algorithms, see, for instance, [7], [8], and [9]. There are two major issues in the application of K-Means-type (non fuzzy or fuzzy) algorithms in cluster analysis. The first issue is that the number of clusters k needs to be determined in advance as an input to these algorithms [3].

In this paper, we propose an agglomerative fuzzy K-Means clustering algorithm for change detection HRCSI images. The new algorithm is an extension to the standard fuzzy K-Means algorithm by introducing a penalty term to the objective function to make the clustering process not sensitive to the initial cluster centres. The new algorithm can produce more consistent clustering results from different sets of initial clusters centres. Combined with cluster validation techniques, the new algorithm can determine the number of clusters in a data set.

2. Image Ingestion

The process of a pair of overlapping multi temporal HRSI images scenes denoted as reference old and target new an input image must have coarse geo-reference information along with Panchromatic and Spectral bans are PAN, RGB,NIR(Near Infra Red), and Texture bands. The ingested over the 'N' number of Scene pair from 'n' image locations. We have capturing the data are Quick Bird NITF raw sensor images, and IKONON images. These data supported by Geospatial Data Abstraction Library (GDAL). If the input image has not been geometrically corrected and RPC data is available, the rectified images significantly improve the registration accuracy. The input images must be in UTM coordinates, all image bands are resample as necessary to the sensor specific nominal panchromatic pixel resolution. The low level sensor resolution spectral-based comparison across the images that have been captured by different sensors, the same sensor Time-Delayed Integration (TDI) level, captured at different times. Pan-sharpening is applied to the resulting multi-spectral bands as needed. Finally to prepare the scene pair for registration we use the reference data to identify overlap between the old and target images.

2.1 Image Registration

Image registration uses a set of extern points (EP) extracted from the PAN band generated from each images, the EP reference matching methods are image sub divided into tiles. The tiles data EP extraction and registration of the image and cross registration are validated algorithm are discuss here. If any image registration error is not accepted and additional matching EPs are extracted the rest of unprocessed data and error registration to be repeated. The registration algorithm, we calculate the 10% cross-validation error (*CVErr*). If all *CVErr* > T, then the image will select the algorithm with low *CVErr* and *CVErr* < T then the selection rules the favour the algorithm. To minimize spectral difference in viewing the geometries, match the reference. The applying the histogram specification to the band histogram use non-vegetated pixels only (NDVI≤ 0.1) to execute the vegetation.

Figure1 - Original Reference Image

Figure1a- Target Image

2.2 Image Extraction

The image extraction is spectral and panchromatic methods most commonly used. We have R, G, B, NIR, PAN, and NDVI feature methods. The Spectral Signatures can vary widely due to differing the geometries or seasons. The feature methods are linear and texture is correlation of pixel along with lines radiating out from the pixel of interest at varying angles. Extract the four linear and texture feature are pixel length, width and azimuth angle and length/width angle. The texture based feature is calculated using Shannon's entropy and skewness measure in 11x11 window [1].

Figure2- Reference Red Color Extraction Image

Figure2a- Target Red Color Extraction Image

3 Image Differences

We calculate a difference feature which reflects the dissimilarity between two corresponding pixels in scenes A and B. The difference feature is corrected to account for small co-registration error by first defining square window of neighbourhood pixel in A and B the size of this window is proportional to the magnitude of co-registration error. Let P and Q be set of pixel values contained within neighbourhood windows cantered at the location of *p* and *q*, respectively. The dissimilarity operator between two pixel value set X and Y, $D(X,Y)=signAbsMin\{d(x,y)\}\forall(x,y) \in X \times Y$, where signAbsMin returns the number (sign intact) whose absolute value is smallest and *d* is a pixel wise dissimilarity operator. The spectral and texture features are

$d(x,y)=y-x$, while for linear feature, we use $d(x_l,x_\theta,y_l,y_\theta)$ $=sign(x_l - y_l)$ $(((x_l V\ y_l) / L)\ A\ ((|x_l - y_l|/L)\ V\ \sin(x_\theta,-y_\theta)|))$. The l and θ subscripts represent the value of the linear feature's pixel length, width and corresponding maximum, minimum length angle, respectively, and L is a normalization factor that takes the maximum value of the pixel length. Thus $d(x_l,x_\theta,y_l,y_\theta)$ calculate the difference for pixel length, width and maximum and minimum length angle feature simultaneously. A *corrected difference* for pixel-level feature is given as δ = max $(D(P, \{q\}),D(\{p\},Q))$. The four linear features are represented using two difference features, whereas a difference feature is calculated for each of the spectral and texture-based features. We also convert the RGB channels to HSV and calculate the HSV and HS distances [10] as two additional difference features, giving us the total of 14 difference features for change detection estimation. Next, we normalize the corrected difference of feature k, δ_k, using feature k scene-pair-wide mean and standard deviation μ_k and σ_k, respectively, by $\hat{\delta}_k = |\ \delta_k-\sigma_k|/\sigma_k$ we calculate the pixel-level change confidence defined as c= $W^{-1}\,\Sigma_k\,w_k$ and $w_k \in[0,1]$ is the weight assigned to the normalized difference. We denote the image whose pixels comprised of change confidence values as the *confidence image* (CI).

4. Change Detection

Let R_k and T_k be the k^{th} feature of the reference and target images, respectively. First, we calculate the "corrected" feature difference for pixel (x,y), $\delta_k(x,y)=R_x(x^*,y^*) - T_k(x,y)$, where (x^*,y^*) is the corrected pixel correspondence that minimizes the difference in feature k, (x^*,y^*) = argmin $(|R_k(x',y') - T_k(x,y)|)$, $(x',y')\in V$ where V is a neighbourhood of pixels cantered at (x,y). This correction accommodates for small local registration errors due to topography, modest building lean from varying sun sensor viewing geometries, etc. The corrected feature difference is normalized as follows: $\hat{\delta}_k=|\delta_k-\mu_k|/\sigma_k$ where σ_k and μ_k are the global mean and standard deviation of δ_k for this scene pair. The confidence image is defined as $C=1/w\Sigma_k w_k\hat{\delta}_k$ where $w_k \in[0,1]$ and W is the sum total of all w_k The pixel confidence $C(x,y)$ represents how confident the algorithm is that significant change occurs at that pixel. The aggregation weights w_k may be used to reflect user confidence of the feature based on prior knowledge. We are now ready to assign a "change" or "no change" label to regions in the confidence image. Threshold is a common approach to a labelling problem like this. However, threshold is an inherently heuristic method of decision-making, and choosing an appropriate threshold is often problematic. To avoid this problem, we introduce a method called the stack filter. We start by defining a stack of threshold values S with threshold increment α, i.e., $\alpha = S_i - S_{i-1}$. S_0 is initialized to a low threshold value. We also create an accumulator image A the same size as C; A is initialized to 0. The stack filters pseudo-code[2].

5. Cluster Validation

The most important parameter in the *K-Means* – type algorithm is the number of clusters. The number of clusters in a data set is a user-defined parameter, which is difficult to specify. In practice, different k values are tried, and the results are compared and analyzed with cluster validation techniques to determine the most appropriate number of cluster. For this purpose, different validation indices have been proposed [19].

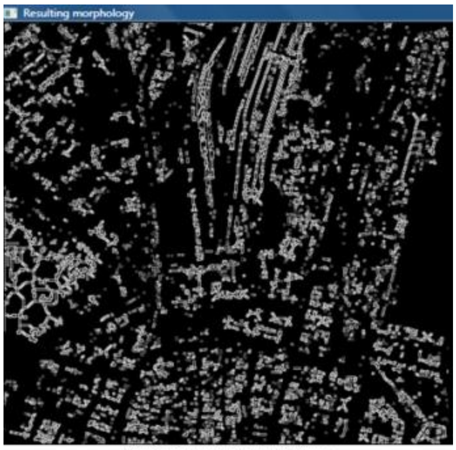


Figure3- Reference Morphology filter Image

Figure1- Original Reference Image

5.1. Agglomerative method

Agglomerative procedures are probably the most widely used of the hierarchical methods. They produce a series of partitions of the data: the first consists of n single member 'clusters'; the last consists of a single group containing all n individuals. The basic operation of all such methods is similar, and will be illustrated for two specific examples, single linkage and centroid linkage. At each stage the methods fuse individuals or groups of individuals which are closest (or most similar). Differences between the methods arise because of the different ways of defining distance (or similarity) between an individual and a group containing several individuals, or between two groups of individuals.

An agglomerative clustering procedure starts with each object as one cluster and forms the nested sequence by successively merging clusters. The main advantage of the agglomerative procedure is that clustering is not influenced by initialization and local minima. In addition, the number of clusters need not be specified a priori. Practitioners can analyze the dendrogram produced by the clustering process, cut the dendrogram at a suitable level, and then identify the clusters. Based on the agglomerative procedure, Frigui and Krishnapuram[19], proposed a new fuzzy clustering algorithm that minimizes an objective function that produces a sequence of partitions with a decreasing number of clusters. Let $X = \{X_1,X_2, X_3,\ldots\ldots\ldots\ldots X_n\}$ be a set of n objects in which each object X_i is represented as $[x_{i,1}, x_{i,2}, x_{i,3},\ldots\ldots\ldots\ldots\ldots x_{i,m}]$, where m is the number of numerical attributes. To cluster X into k clusters by the agglomerative fuzzy *K-Means* algorithm is to minimize the following objective function

$$P(U,Z) = \sum_{j=1}^{m} \sum_{j=1}^{n} u_{ij}D_{ij}+\lambda\sum_{j=1}^{m} \sum_{j=1}^{n} u_{ij} \log u_{ij} \quad \text{----(1)}$$

Subject to $\sum_{j=1}^{n} u_{i,j}=1, \quad u_{i,j} \in [0,1], \quad 1 \le i \le n,$ .... (2)

Where U = $[u_{i,j}]$ is an *n-by-k* partition matrix, $u_{i,j}$ represents the association degree of membership of the i^{th} object x_i j^{th} cluster z_j, $Z = [z_1, z_2, z_3, \ldots\ldots\ldots z_k]^T$ is an *k-by-m* matrix containing the cluster centres, and $D_{i,j}$ is a dissimilarity measure between the j^{th} cluster centre and the i^{th} object. Here, the square of the Euclidean norm is used as the dissimilarity measure, i.e.,

$$D_{i,j} = \sum_{i=1}^{m} (z_{j,l} - x_{i,l})^2$$

Such dissimilarity measure is commonly used in clustering. The first term in the cost function of the standard *K*-means algorithm. The second term is added to maximize the negative object-to-clusters membership entropy in the clustering process.

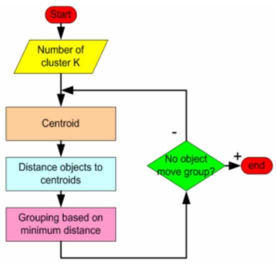
6. Fuzzy *K*-means

The fuzzy *k*-means algorithm (Bezdek, 1974b) is an extension of the *k*-means algorithm for fuzzy clustering. Give a data set $D = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, the algorithm is based on minimization of the objective function

$$J_q(U.V) = \sum_{j=1}^{n} \sum_{j=1}^{k} u^2_{ij} d^2(x_j, V_i) \quad \text{------(4)}$$

with respect to U (a fuzzy k-partition of the data set) and to V (a set of k prototypes), where q is a real number greater than 1, V_i is the centroid of cluster i, u_{ij} is the degree of membership of object $\mathbf{x}_j$ belonging to cluster i, $d^2(\cdot,\cdot)$ is an inner product metric, and k is the number of clusters. The parameter q controls the "fuzziness" of the resulting clusters. The parameter q controls the "fuzziness" of the resulting clusters (Bezdek, 1981).

6.1 Fuzzy *K-means* flow diagram



6.2 Fuzzy *K-means* algorithm

Step1: Choose initial centroids V_i $(i = 1,2,\ldots\ldots\ldots k)$;

Step2: Compute the membership matrix as follows:

$i = 1,2,3\ldots\ldots\ldots\ldots k,\ j= 1,2,3,\ldots\ldots\ldots\ldots\ldots n;$ ....(5)

$$u_{ij}=[d^2(x_j,v_i]^{-1/q-1} / \sum_{j=1}^{m} [d^2(x_j,v_l)]^{-1/q-1}$$

Step3: Compute new centroids V_i $(i = 1,2,3\ldots\ldots, k)$ as

$$V_i=\sum_{j=1}^{n} u^q_{ij}X_j / \sum_{j=1}^{n} u^q_{ij}$$

and update the membership matrix (u_{ij}) to (u_{ij}) according to equation (5).

Step4: If Max_{ij} then stop; otherwise go to step Step3, where ∈ is a termination criterion between 0 and 1.

The fuzzy clustering is carried out via an iterative optimization of equation (4). The procedure of the optimization is shown in algorithm.

The hyper ellipsoidal clusters and clusters with variable densities and unequal sizes, (Gath and Geva(1989)) presented and "exponential" distance measure based on maximum likelihood estimation, i.e., where *Fi* is the fuzzy covariance matrix of the i^{th} cluster and P_i is the *a priori* probability of selecting the i^{th} cluster.

The above distance is used in the calculation of $h(i|x_j)$, the probability of selecting the i^{th} cluster given the j^{th} object:

$$h(i|xj)= 1/d^2(x_j,V_i) / \sum_{l=1}^{k} (d^2_e(x_j,V_l) \quad \ldots\ldots\ldots\ldots\ldots(6)$$

$$(d^2_e(x_j,V_i) = \sqrt{\det(Fi)} / P_i \exp[(x_j-V_i)^T F^{-1}_i(x_j-V_i)/2]$$

If we let q=2 in equation (5), $h(i|x_jI)$ defined in equation (6) is similar to u_{ij}. Thus, if we substitute equation (6) instead of equation (5) in step 2 of Algorithm the fuzzy k-means algorithm becomes the FMLE (Fuzzy modification of the Maximum Likelihood Estimation) algorithm. In addition to computing the centroids, Step 3 of need to calculate P_i and F_i[12] .

$$P_i=1/n \sum_{j=1}^{n} h(i|x_j),$$

$$F_i=\sum_{j=1}^{n} h(i|x_j)(x_j-V_i)(x_j-V_i)^T / \sum_{j=1}^{n} h(i|x_j)$$

6.3 Implementation

The image registration, ingestion, future extraction and change detection in High resolution Satellite Imagery are implemented in IDL and utilize various built in function in IDL [20].

7. Conclusion and future work

We proposed an *AK-means* clustering algorithm called agglomerative *K-means* to perform unsupervised clustering of change detection results automatically produced by the IDL system. The *AK-means* algorithm utilizes the regular FCM algorithm to stabilize cluster partition. *AK-means* algorithm such that it can estimate the optimal cluster number as it runs through its alternating – optimization process. We enhance this capability by adding an ability to find the optimal cluster number within a specific range using cluster validity measures as indicators. We presented the application of *AK-means* to find the cluster in the IDL change detection results. Future work is moving object changes detection using clustering algorithm.

8. REFERENCES

[1] IEEE Transactions on Geosciences and Remote Sensing, Clustering of Detected Changes in High-Resolution Satellite Imagery Using a Stabilized Competitive Agglomeration Algorithm, Ozy Sjahputera, J.Scott, C.Claywell, N.Klaric, J.Hudson, M.Keller and H.Davis.

[2] M. Datcu and K. Seidel, "Human-cantered concepts for exploration and understanding of Earth observation images," IEEE Trans. Geosci. Remote Sens., vol. 43, no. 3, pp. 601–609, Mar. 2005.

[3] Agglomerative Fuzzy K-Means Clustering Algorithm with Selection of Number of Clusters, Mark Junjie Li, Michael K. Ng, Yiu-ming Cheung, Senior Member, IEEE, and Joshua Zhexue Huang

[4] D. Brunner, G. Lemoine, F. X. Thoorens, and L. Bruzzone, "Distributed geospatial data processing functionality to support collaborative and rapid emergency response," IEEE J. Sel. Topics Appl. Earth Obs. Remote Sens., vol. 2, no. 1, pp. 33–46, Mar. 2009.

[5] M. Tyagi, F. Bovolo, A. K. Mehra, S. Chaudhuri, and L. Bruzzone, "A context-sensitive clustering technique based on graph-cut initialization and expectation-maximization algorithm," IEEE Geosci. Remote Sens. Lett., vol. 5, no. 1, pp. 21–25, Jan. 2008.

[6] U. Maulik and I. Saha, "Automatic fuzzy clustering using modified differential evolution for image classification," IEEE Trans. Geosci. Remote Sens., vol. 48, no. 9, pp. 3503–3510, Sep. 2010.

[7] A. Mukhopadhyay and U. Maulik, "Unsupervised pixel classification in satellite imagery using multiobjective fuzzy clustering combined with SVM classifier," IEEE Trans. Geosci. Remote Sens., vol. 47, no. 4, pp. 1132–1138, Apr. 2009.

[8] C. Yang, L. Bruzzone, F. Sun, L. Lu, R. Guan, and Y. Liang, "A fuzzy-statistics-based affinity propagation technique for clustering in multispectral images," IEEE Trans. Geosci. Remote Sens., vol. 48, no. 6, pp. 2647–2659, Jun. 2010.

[9] T. Celik, "Unsupervised change detection in satellite images using principal component analysis and k-means clustering," IEEE Geosci. Remote Sens. Lett., vol. 6, no. 4, pp. 772–776, Oct. 2009.

[10] S. Ghosh, N. S. Mishra, and A. Ghosh, "Unsupervised change detection of remotely sensed images using fuzzy clustering," in Proc. Int. Conf. Adv. Pattern Recog., 2009, pp. 385–388.

[11] M. J. Carlotto, "A cluster-based approach for detecting man-made objects and changes in imagery," IEEE Trans. Geosci. Remote Sens., vol. 6, no. 1, pp. 189–202, Feb. 2005.

[12] Data Clustering Theory, Algorithms, and Applications, Guojun Gan York University Toronto, Ontario, Canada

[12] Cluster Analysis 5th Edition Brian S. Everitt . Sabine Landau Morven Leese . Daniel Stahl King's College London, UK

[13] fuzzy algorithms with application to image processing and pattern recognition

[14] Fuzzy Cluster Analysis -John Wiley and Sons

[15] FUZZY LOGIC WITH ENGINEERING APPLICATIONS Third Edition Timothy J. Ross *University of New Mexico, USA*

[16] FUZZY MODELS AND ALGORITHMS FOR PATTERN RECOGNITION AND IMAGE PROCESSING

[17] Theories, Methods, and Applications Remote Sensing and GIS Integration Qihao Weng, Ph.D.

[18] SATELLITE REMOTE SENSING FOR ARCHAEOLOGY *Sarah H. Parcak*

[19] A.K. Jain and R.C. Dubes, Algorithms for Clustering Data. Prentice, Hall, 1988.

[20] Image.Processing.Handbook.6th.Edition.Apr.2011 CRC.Press

[21] IDL Reference Guide