

SPECIAL PERTURBATIONS UNCORRELATED TRACK PROCESSING

James G. Miller*

Two historical uncorrelated track (UCT) processing approaches have been employed using general perturbations (GP) orbit determination theory. The Cuthbert-Morris algorithm clusters UCTs based on plane, drift rate of the right ascension of the ascending node, and period matching. A pattern recognition tool developed by Lockheed Martin finds patterns in the trends of GP UCT element set parameters over time. A new special perturbations (SP) hierarchical agglomerative clustering algorithm and SP track-oriented multiple hypothesis tracking (MHT) algorithm are considered for SP UCT processing. Both SP UCT processing algorithms show improved performance over the GP UCT processing algorithms for a stressing test case.

INTRODUCTION

Two historical uncorrelated track (UCT) processing approaches have been employed using general perturbations (GP) orbit determination theory. The Cuthbert-Morris (C-M) algorithm clusters UCTs based on plane, drift rate of the right ascension of the ascending node, and period matching. One-track GP element sets are created from each UCT, and a plot program allows the user to visually see trends in the UCT element set parameters and to manually cluster UCTs together that potentially belong to the same object. A pattern recognition tool developed by Lockheed Martin¹ attempts to automate this manual process that the human eye performs from the plots of the UCT element set parameters. A stressing test case for UCT processing is used to show the limitations of these GP approaches to UCT processing.

Two new special perturbations (SP) UCT processing algorithms are considered and evaluated against this stressing case. The first SP UCT processing algorithm is a hierarchical agglomerative clustering algorithm. Two clustering methodologies are considered, single-link and complete-link, which represent the two extremes of hierarchical agglomerative clustering methodologies. The hierarchical agglomerative clustering algorithm with the single-link clustering methodology tends to form elongated clusters, whereas more spherical clusters are formed with the complete-link clustering methodology. The hierarchical agglomerative clustering algorithm performs better with the complete-link clustering methodology than the single-link clustering methodology on the stressing case. The second SP UCT processing algorithm is an SP track-oriented multiple hypothesis tracking (MHT) algorithm. Multiple tracks are updated with the same UCT and maintained in a tree structure. Multiple hypotheses are formulated, followed by global-level track pruning. Both SP UCT processing algorithms show improved performance over the GP UCT processing algorithms for this stressing test case.

TEST CASE

* The MITRE Corporation, 1155 Academy Park Loop, Colorado Springs, CO 80910-3704, USA.

The stressing test case consists of three satellites, whose orbits are very close to each other. The satellites are designated as objects 1, 2 and 3. The term satellite is used in the generic sense to refer to any man-made object in orbit around the earth, including debris objects. Figure 1 shows a plot of the operational GP element set parameters of these three satellites from day 212 to day 222 of year 2006. These satellites are very small debris objects and the tracking data is very sparse, particularly for object 3. Each track consists of three observations separated by 10 seconds. The element sets are produced by a batch GP differential correction (DC) with an orbit determination interval of several days so that multiple tracks are used in each DC. When a new track is received, the element set is updated with the new track of observations and all the observations in the sliding orbit determination interval. These satellites represent the kind of objects that show up as UCTs, and in fact, were initially tracked as UCTs before they were cataloged. There are linear trends in some of the element set parameters with good separation between the values. For example, the period of object 1 is about 101.51 minutes, the period of object 2 is about 101.52 minutes, and the period of object 3 is about 101.55 minutes. The drift rates of the right ascension of the ascending node (RANODE) are the same for the three satellites, but the right ascension of the ascending nodes are slightly offset from one another for a given time. The inclinations of objects 1 and 2 are the same, but the inclination of object 3 is slightly larger than the inclination of the other two satellites.

Figure 2 is a subset of Figure 1, where three element sets were chosen for each satellite. The epoch time between element sets is several days, which will stress the UCT processing algorithms. The linear trends in some of the element set parameters are still evident in Figure 2 with good separation between the values.

Figure 3 shows a plot of one-track GP element sets for the same satellites in Figure 2. Each element set is obtained by first performing an initial orbit determination using the Herrick-Gibbs² algorithm for each track of observations followed by a Simplified General Perturbations 4 (SGP4) DC to refine the element set. Because each GP element set in Figure 3 is created from only one track of observations, there are more errors in the element set parameters in Figure 3 than in Figure 2. The linear trends in Figure 2 are no longer present in Figure 3, and the values of the element set parameters can no longer be separated by satellite. The test case consists of the tracks that are used in Figure 3, where the identity of the satellite for each track is hidden from the UCT processing algorithm. The satellite tag on the observations of each track is changed to a 9xxxx satellite number so that each track has a different set of 9xxxx numbers. This retagging of the observations from the original satellite numbers to 9xxxx numbers simulates UCTs and provides a test case where the true identity of the UCTs is known.

The Cuthbert-Morris algorithm is applied to this test case, but it produces just one cluster of three tracks corresponding to object 2. This is not surprising given that the algorithm is GP-based, and the element set parameter values are not well separated by satellite. The Lockheed Martin pattern recognition tool is also applied to this test case, but it produces no clusters. These GP-based algorithms are limited in handling this stressing test case.

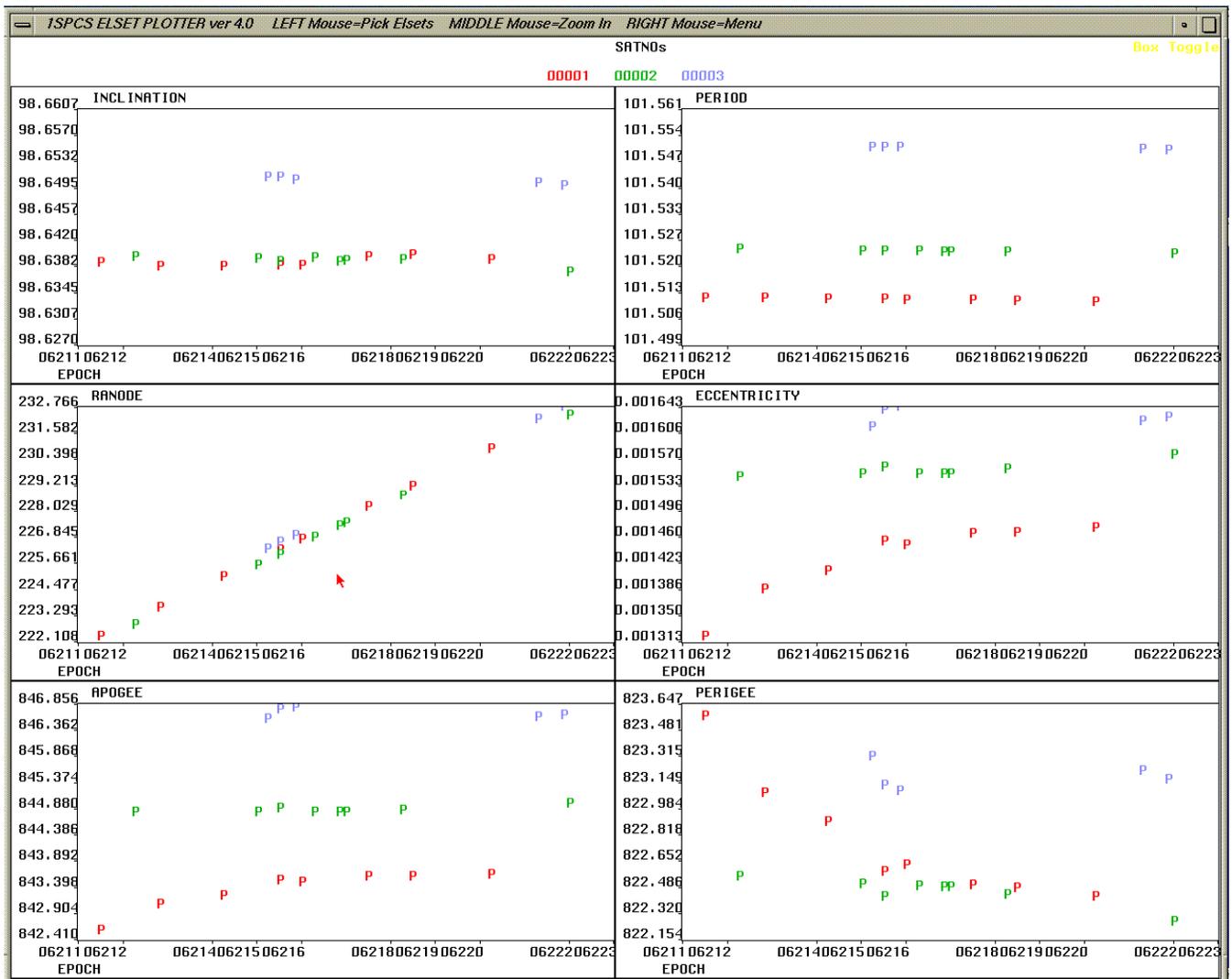


Figure 1 Operational GP Element Sets



Figure 2 Subset of Operational GP Element Sets



Figure 3 One-Track GP Element Sets

CLASSIFICATION ALGORITHMS

Classification algorithms are applicable to the UCT processing problem. Classification algorithms³ form a broad set of algorithms that have been applied to a diverse set fields, including signal processing and analysis, target discrimination, pattern recognition, biological taxonomy, and database mining, to mention a few. Figure 4 shows a hierarchy of classification algorithms, which is not complete but only intended to illustrate the hierarchy. The classification terminology among the various fields is not consistent, and the terms used in Figure 4 may not agree with what may be used in a particular field. The first division indicates whether each object to be classified can be in only one group (exclusive) or in multiple groups (nonexclusive). The Cuthbert-Morris algorithm is a heuristic algorithm that is nonexclusive. Multiple hypothesis tracking algorithms are also nonexclusive. The exclusive classification algorithms can be divided into unsupervised learning and supervised learning. The Lockheed Martin pattern recognition tool uses a supervised learning adaptive resonance theory (ART) algorithm, which mimics the human brain. The Lockheed Martin pattern recognition tool must be trained on a data set for which the right answers are known.

Clustering algorithms⁴ can be divided into hierarchical and partitional. The k-means algorithm is a very popular clustering algorithm, but it requires *a priori* knowledge of the number of clusters. For the UCT processing problem, the number of real clusters (satellites) is unknown, so this algorithm is not suitable for this problem. The hierarchical clustering algorithms can be divided into agglomerative and divisive. The agglomerative algorithms work from the bottom up and start with every object in its own cluster. The agglomerative algorithms combine clusters based on a proximity measure. The divisive algorithms work from the top down and start with all objects in one cluster and iteratively divide clusters into smaller and smaller clusters. The algorithms considered here for SP UCT processing are hierarchical agglomerative clustering and MHT.

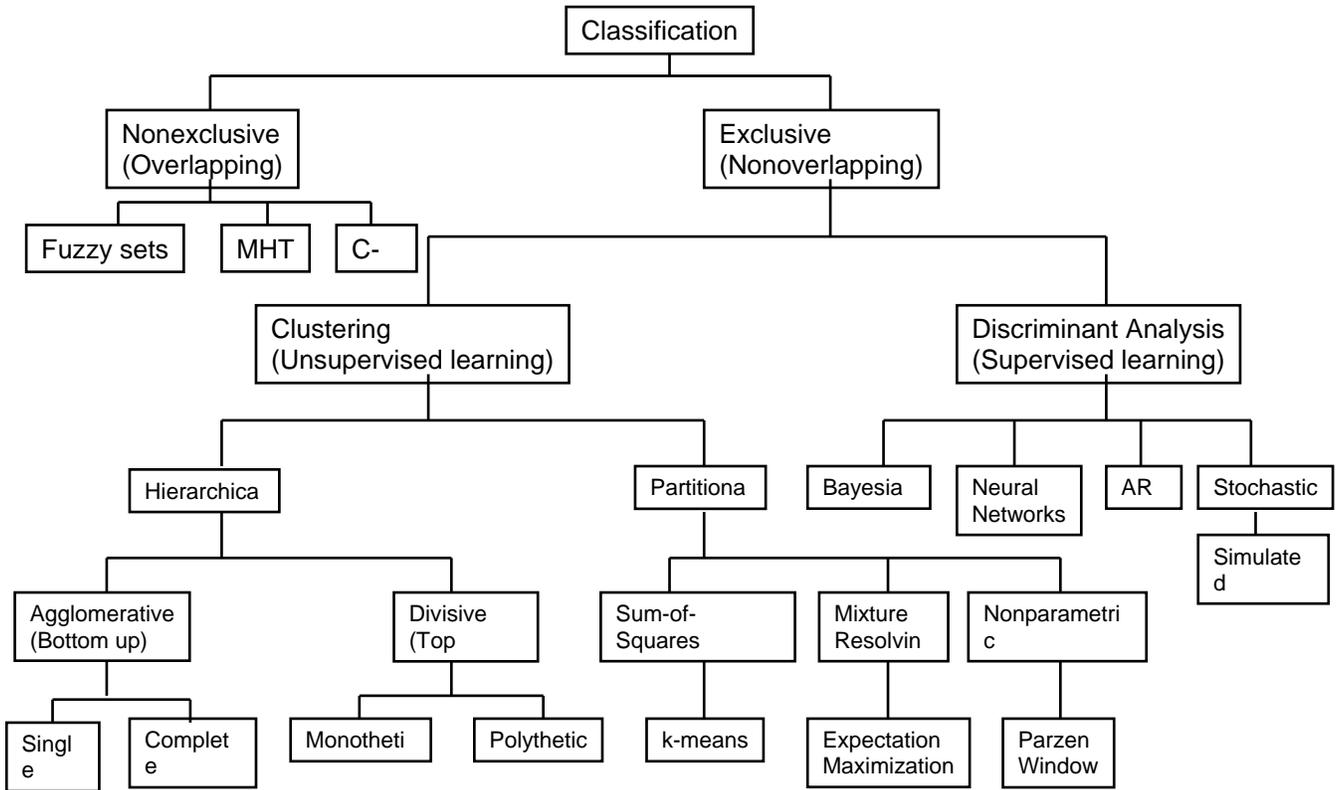


Figure 4 Hierarchy of Classification Algorithms

HIERARCHICAL AGGLOMERATIVE CLUSTERING

Clustering algorithms⁴ have four components, pattern representation, proximity measure, clustering methodology, and cluster validation. A simple example will illustrate the hierarchical agglomerative clustering algorithms. Figure 5 shows four points on the real number line. The pattern representation for this simple example is a real number. The proximity measure is the distance between two real numbers. Two clustering methodologies are considered, single-link and complete-link. At each stage of the hierarchy, the single-link clustering methodology combines the two closest clusters, where the distance between two clusters is the minimum of the distance between all pairs of points from the two clusters. At each stage of the hierarchy, the complete-link clustering methodology combines the two closest clusters, where the distance between two clusters is the maximum of the distance between all pairs of points from the two clusters.

Figure 5 shows dendrograms for the single-link and complete-link clustering methodology. Both

methodologies start with all four points in their own cluster. The single-link clustering methodology first combines points labeled (1) and (2) into a new cluster because these two points are the closest with a distance of 2, which is the cluster proximity value on the dendrogram. At this stage there are three clusters, (1) and (2), (3), and (4). The next step is to compute the pair-wise distance between these three clusters. According to the single-link clustering methodology, the distance between cluster (1) and (2) and the singleton cluster (3) is 3. The distance between cluster (1) and (2) and the singleton cluster (4) is 7. The distance between the singleton cluster (3) and the singleton cluster (4) is 4. The minimum distance between the clusters is 3, which is the cluster proximity value, and cluster (1) and (2) is combined with the singleton cluster (3). At this stage there are two clusters, (1), (2), (3) and the singleton cluster (4). The last step combines these two clusters into one cluster of all four points with cluster proximity value 4.

The complete-link clustering methodology also first combines points labeled (1) and (2) into a new cluster. At this stage there are the same three clusters as in the single-link clustering methodology. According to the complete-link clustering methodology, the distance between cluster (1) and (2) and the singleton cluster (3) is 5. The distance between cluster (1) and (2) and the singleton cluster (4) is 9. The distance between the singleton cluster (3) and the single cluster (4) is 4. The minimum distance between the clusters is 4, which is the cluster proximity value, and singleton cluster (3) and singleton cluster (4) are combined into a new cluster. At this stage there are two clusters, (1) and (2), and (3) and (4). The complete-link clustering methodology gives a different set of clusters at this stage than the single-link clustering methodology. The last step combines these two clusters into one cluster of all four points with cluster proximity value 9.

The hierarchy can be cut at any proximity value on the dendrogram to form a set of clusters with the number of vertical lines traversed corresponding to the number of clusters. For this example, there is no right answer since the data was invented to illustrate the difference between the single-link and complete-link clustering methodologies. For real-world data, there needs to be a way to determine where to cut the hierarchy somewhere between every object in its own cluster and all objects in one cluster. Plotting the cluster proximity values versus the number of clusters formed gives insight into where to cut the hierarchy. The slope of the plot of proximity values is constant for the single-link case in Figure 6. Therefore, there is no natural place to cut the hierarchy for the single-link case. The slope of the plot of proximity values increases significantly from two clusters to one cluster in Figure 7. Therefore, the hierarchy can be cut at two clusters in a reasonable sense for the complete-link case.

For SP UCT processing, the pattern representation will be the SP state vectors created from UCTs. An initial orbit determination is performed for each UCT using the Herrick-Gibbs algorithm, followed by an SP DC to create an SP state vector with covariance matrix. The proximity measure is a distance (or distance squared) function between two SP vectors. A possible proximity measure is the distance squared function given by

$$(1) \quad d^2(x_1, x_2) = (x_2 - x_1)^T (C_1 + C_2)^{-1} (x_2 - x_1),$$

where x_1 and x_2 are SP state vectors propagated to the midpoint of the epoch times, C_1 and C_2 are their propagated covariance matrices, respectively, and T denotes the transpose of the 6-dimensional column state vector. The distance squared function is a Chi squared statistic with six degrees of freedom.

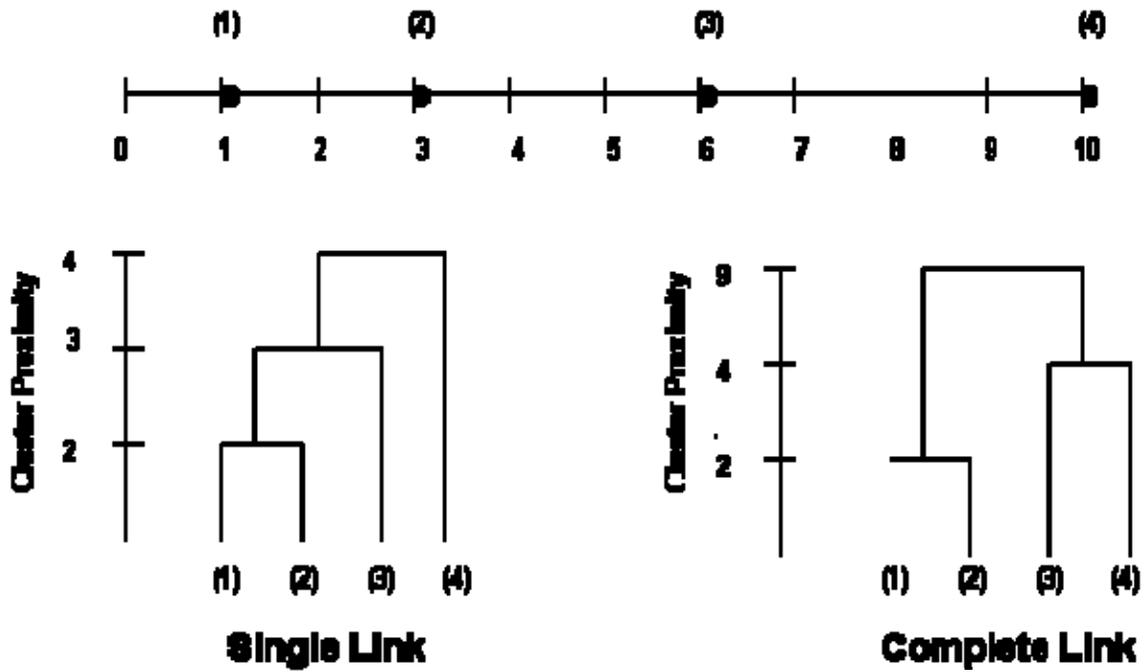


Figure 5 Single-Link and Complete-Link Clustering Methodologies

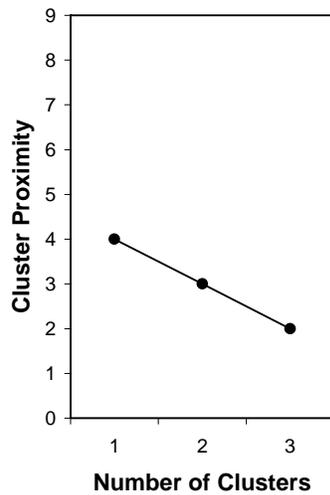


Figure 6 Single-Link Cluster Proximity Values Versus Number of Clusters

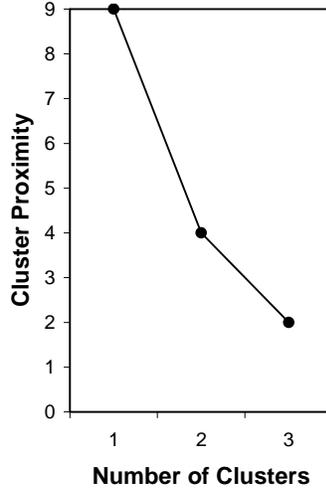


Figure 7 Complete-Link Cluster Proximity Values Versus Number of Clusters

It turns out that the propagated covariance matrix associated with a one-track SP state vector does not necessarily remain positive definite, and therefore the distance squared function in Eq. (1) can be negative. Thus, Eq. (1) is not a suitable proximity measure. Instead,

$$(2) \quad d^2(x_1, x_2) = (r_2 - r_1)^T (R_1 + R_2)^{-1} (r_2 - r_1) + (v_2 - v_1)^T (V_1 + V_2)^{-1} (v_2 - v_1)$$

will be used as the proximity measure since it remains positive, where r denotes the positional part of the state vector, R the denotes the 3 x 3 positional part of the covariance matrix, v denotes the velocity part of the state vector, and V denotes the 3 x 3 velocity part of the covariance matrix.

Using the proximity measure in Eq. (2) with the single-link clustering methodology yields the hierarchical clustering given in Table 1. The track numbers are color coded to agree with the color of the satellite numbers in Figure 3. The algorithm has no knowledge of the association of track number to satellite number. The plot of the cluster proximity values versus the number of clusters is given in Figure 8. There is no natural place to cut the hierarchy based on the slopes in Figure 8. The correct answer is 3 clusters, but column 3CL in Table 1 does not cluster the tracks based on the color of the track numbers. Column 3CL clusters tracks 1, 2, 3, 4, 6, 7, and 9 together, and track 5 is a singleton cluster, and track 8 is a singleton cluster. Using the proximity measure in Eq. (2) with the complete-link clustering methodology yields the hierarchical clustering given in Table 2. The plot of the cluster proximity values versus the number of clusters is given in Figure 9.

Table 1

SINGLE-LINK HIERARCHICAL CLUSTERING WITH PROXIMITY MEASURE EQ. (2)

TRK NO.	2CL	3CL	4CL	5CL	6CL	7CL	8CL	9CL
1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	2
3	1	1	1	3	3	3	3	3
4	1	1	1	1	4	4	4	4
5	5	5	5	5	5	5	5	5
6	1	1	6	6	6	6	6	6
7	1	1	1	1	1	7	7	7
8	1	8	8	8	8	8	8	8
9	1	1	1	1	1	1	9	9

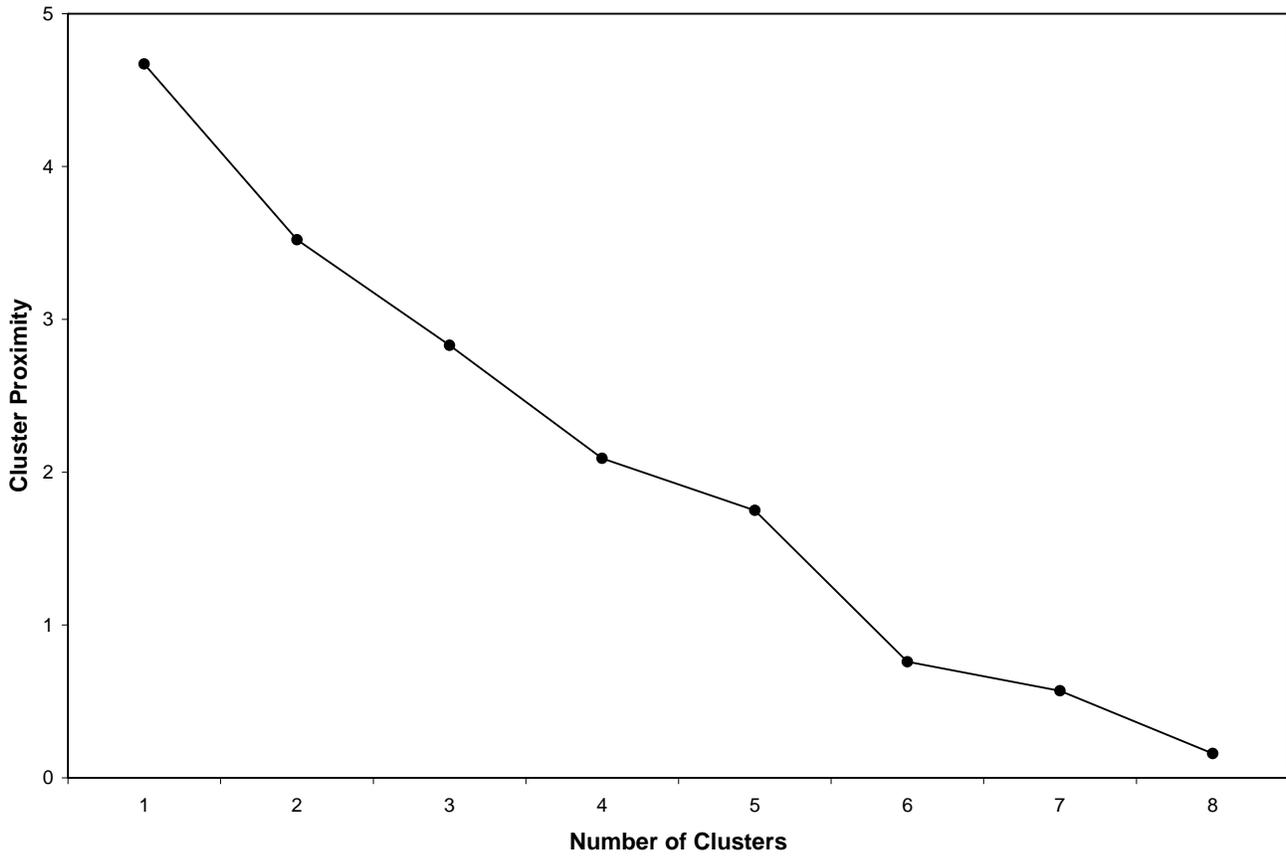


Figure 8 Single-Link Cluster Proximity Values Versus Number of Clusters with Proximity Measure in Eq. (2)

**Table 2
COMPLETE-LINK HIERARCHICAL CLUSTERING WITH PROXIMITY MEASURE EQ. (2)**

TRK NO.	2CL	3CL	4CL	5CL	6CL	7CL	8CL	9CL
1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	2
3	3	3	3	3	3	3	3	3
4	3	3	3	3	3	4	4	4
5	1	1	5	5	5	5	5	5
6	1	1	5	5	6	6	6	6
7	3	7	7	7	7	7	7	7
8	3	7	7	8	8	8	8	8
9	3	7	7	7	7	7	9	9

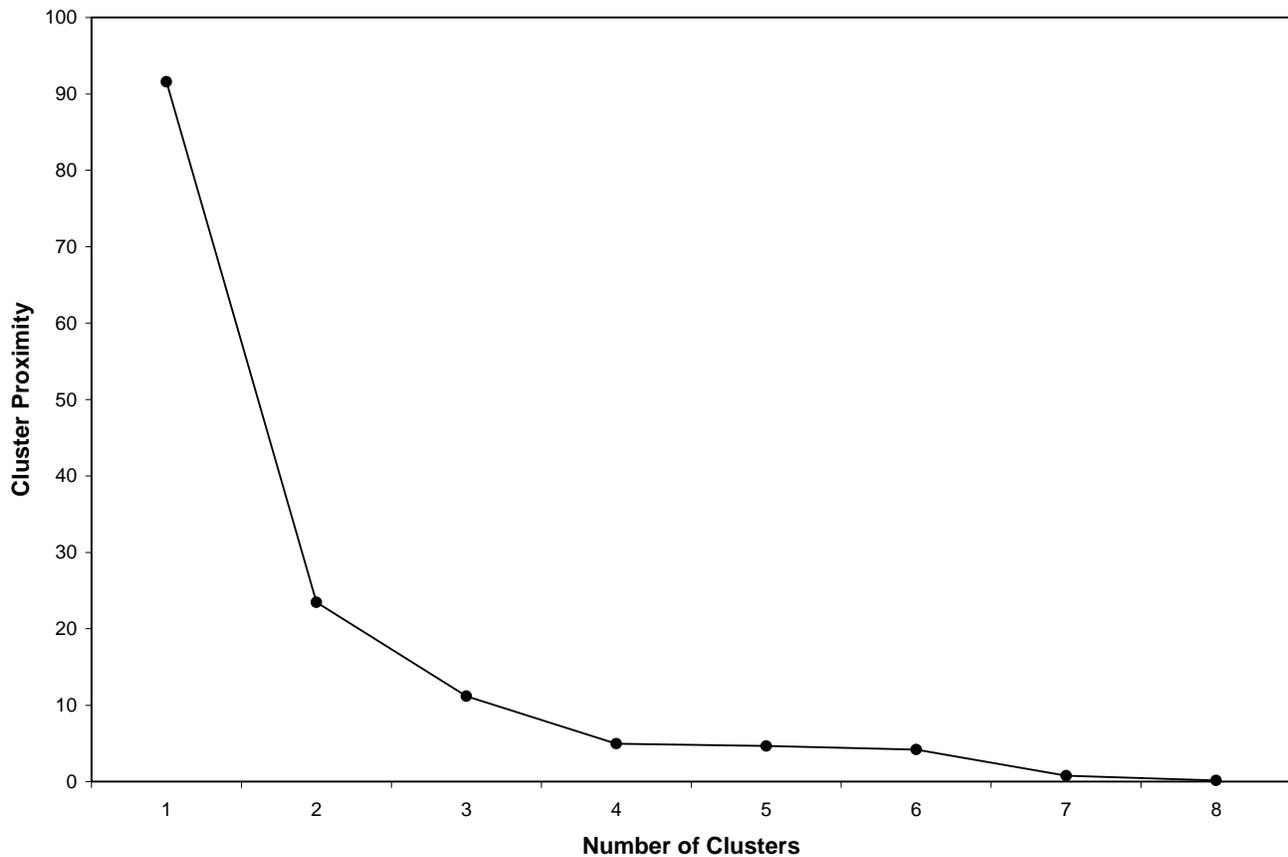


Figure 9 Complete-Link Cluster Proximity Values Versus Number of Clusters with Proximity Measure in Eq. (2)

The hierarchy could be cut at 2 or 4 clusters based on Figure 9, neither of which is the correct answer. The correct answer is 3 clusters, but column 3CL in Table 2 does not cluster the tracks based on the color of the track numbers. However, the clustering in column 3CL for the complete-link methodology looks better than the clustering in column 3CL for the single-link methodology. The first error made in Table 2 is in column 6CL when track numbers 3 and 4 are clustered together. Once an error is made in hierarchical agglomerative clustering, it propagates up the hierarchy. This is one of the weaknesses of hierarchical agglomerative clustering.

The covariance obtained from one track of observations may not be very reliable, particularly when it is propagated for several days. The problem with hierarchical agglomerative clustering may not be with the algorithm, but rather with the proximity measure used. As an alternative to Eq. (2), consider

$$(3) \quad d(x_1, x_2) = \text{RMS},$$

where RMS is the weighted root mean square from the SP DC using the two tracks of observations that created SP state vectors x_1 and x_2 . Table 3 shows the proximity matrix obtained from Eq. (3). The matrix is symmetric with zeros on the diagonal. The hierarchical agglomerative clustering algorithm only works with the upper triangular part of the proximity matrix, so only that part of the matrix is displayed in Table 3.

Table 3
PROXIMITY MATRIX FOR EQ. (3)

Track No.	1	2	3	4	5	6	7	8	9
1		1.0025	1.3369	9.4411	2.1512	6.5489	8.9034	5.4633	1.0377
2			1.4738	1.5713	4.2136	3.3074	3.0339	2.8473	4.5877
3				3.8946	4.1595	7.5899	5.3098	9.8633	1.6652
4					0.9479	1.3976	21.1519	10.7266	2.6254
5						1.4924	10.4098	29.6113	4.1377
6							14.2618	18.4321	3.5756
7								0.9586	0.7473
8									0.9693
9									

Using the proximity measure in Eq. (3) with the single-link clustering methodology yields the hierarchical clustering given in Table 4. The plot of the cluster proximity values versus the number of clusters is given in Figure 10. There is no natural place to cut the hierarchy based on the slopes in Figure 10. The correct answer is 3 clusters, but column 3CL in Table 4 does not cluster the tracks based on the color of the track numbers. The first mistake is made in column 4CL for 4 clusters. The cause of the error is the small value in row 1 and column 9 of Table 3.

Table 4
SINGLE-LINK HIERARCHICAL CLUSTERING WITH PROXIMITY MEASURE EQ. (3)

TRK NO.	2CL	3CL	4CL	5CL	6CL	7CL	8CL	9CL
1	1	1	1	1	1	1	1	1
2	1	1	1	1	2	2	2	2
3	1	1	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4
5	4	4	4	4	4	4	5	5
6	4	6	6	6	6	6	6	6
7	1	1	1	7	7	7	7	7
8	1	1	1	7	7	8	8	8
9	1	1	1	7	7	7	7	9

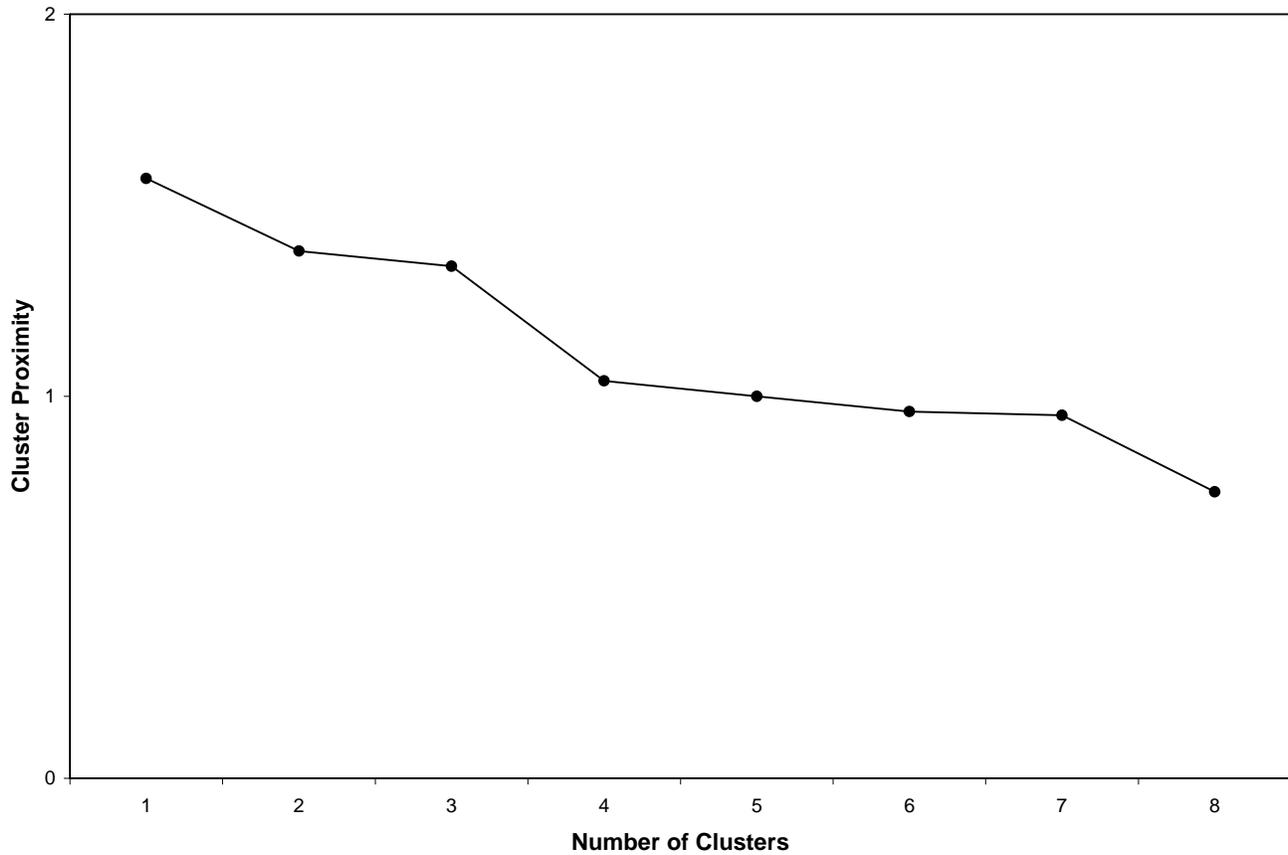


Figure 10 Single-Link Cluster Proximity Values Versus Number of Clusters with Proximity Measure in Eq. (3)

Using the proximity measure in Eq. (3) with the complete-link clustering methodology yields the hierarchical clustering given in Table 5. The plot of the cluster proximity values versus the number of clusters is given in Figure 11. The hierarchy should be cut at 3 clusters based on Figure 11, which is the correct number of clusters. Also, column 3CL in Table 5 clusters the tracks to the correct satellites.

**Table 5
COMPLETE-LINK HIERARCHICAL CLUSTERING WITH PROXIMITY MEASURE EQ. (3)**

TRK NO.	2CL	3CL	4CL	5CL	6CL	7CL	8CL	9CL
1	1	1	1	1	1	1	1	1
2	1	1	1	1	2	2	2	2
3	1	1	1	3	3	3	3	3
4	1	4	4	4	4	4	4	4
5	1	4	4	4	4	4	5	5
6	1	4	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7
8	7	7	7	7	7	8	8	8
9	7	7	7	7	7	7	7	9

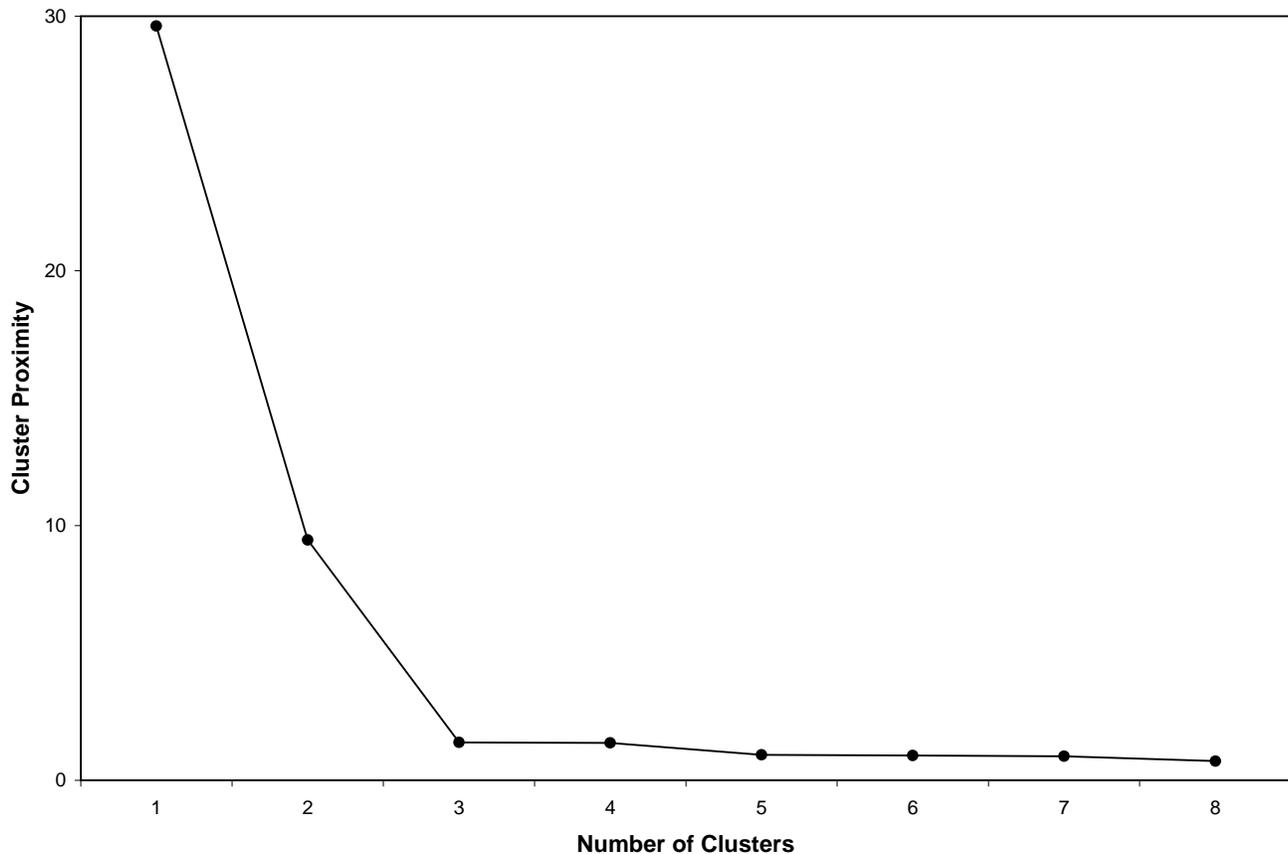


Figure 11 Complete-Link Cluster Proximity Values Versus Number of Clusters with Proximity Measure in Eq. (3)

MULTIPLE HYPOTHESIS TRACKING

Multiple hypothesis tracking⁵ usually considers the concept of a scan of observations and an observation-to-track association problem. The concept of a scan of observations is not relevant to the UCT processing problem. A scan interval of several days needs to be considered for the UCT processing problem instead of a scan of observations over a short period of time. Also, a track-to-track association problem will be considered instead of an observation-to-track association problem. It is assumed that all the observations in a UCT belong to the same satellite. This is a fairly good assumption since space is very big and the density of satellites is very small. We do not really have a multi-target tracking problem in the traditional sense with an observation-to-track association problem. So for each new UCT received, the one-track SP state vector is created by the process described above. The track-to-track association problem is to determine how to associate a new UCT with any previously received UCTs. The concept of gating can be applied, and MHT considers all possible associations that fall within some gating threshold. Something similar to Eq. (1) or Eq. (2) could be considered for the gating criterion, but these equations did not perform well for the hierarchical agglomerative clustering problem. Something similar to Eq. (1) may be suitable for later stages in the MHT problem after several UCTs have been strung together in a longer track, but initially Eq. (3) will be used for the gating criterion. The threshold for gating needs to be large enough to not exclude any true track-to-track associations, and not so large as to include too many false track-to-track associations.

All the values for Eq. (3) for the test case, which simulates the UCT processing problem, are

contained in Table 3. An RMS value of 3.0 will be chosen as the gating threshold, which includes 6 false track-to-track associations, namely (1, 5), (1, 9), (2, 4), (2, 8), (3, 9), and (4, 9). The time order of the UCTs for the test case is 4, 1, 7, 5, 6, 2, 3, 8, and 9. Using Eq. (3) as the gating criterion with threshold 3.0, we obtain the tree structure of associated UCTs in Figure 12. There are seven separate trees with head nodes 4, 1, 7, 5, 2, 3, and 8.



Figure 12 Tree Structure of Associated UCTs

A pruning algorithm is needed to reduce the number of hypotheses represented by the tree structure in Figure 12. First, prune all leaves in the trees that have a depth of 2 (only two UCTs). Next, compute the SP DC for all leaves with at least a depth of 3, using only the first three UCTs if the depth is more than 3. The RMS for these SP DCs is given in Table 6.

**Table 6
MHT RMS**

UCTs	RMS
4, 5, 6	1.5575
4, 2, 3	134.4647
4, 2, 8	157.4187
1, 5, 6	102.4522
1, 2, 3	1.9562
1, 2, 8	295.8171
1, 3, 9	671.3066
7, 8, 9	1.5503
2, 3, 9	271.6341
2, 8, 9	251.3571

From Table 6, the only leaves of the MHT trees that survive are the right answers, namely, 4, 5, 6 corresponding to object 2; 1, 2, 3 corresponding to object 1; and 7, 8, 9 corresponding to object 3.

CONCLUSIONS

GP UCT processing algorithms are limited in handling the stressing test case considered here. Two SP UCT processing algorithms, hierarchical agglomerative clustering, using the complete-link clustering methodology with the SP DC RMS as the proximity measure, and MHT with the SP DC RMS as the gating criterion, associate the UCTs in the test case with the correct satellites. In general, it is very difficult to validate clustering algorithms, and there are no established validation methodologies. All clustering algorithms will group data into clusters, but the clusters may not represent any real or natural grouping of the data. The validation of the clustering algorithm used here relies on a blind test in which the correct clusters are known. These SP UCT processing algorithms need to be tested on more test cases in which the correct answers are known to validate the algorithms. The algorithms then need to be tested on true UCTs to validate the algorithms against real-world data.

REFERENCES

1. Lockheed Martin Mission Systems, *UCT Pattern Recognition User's Manual*, December 31, 2004.
2. D. A. Vallado, *Fundamentals of Astrodynamics and Applications*, 2nd edition, McGraw-Hill, Inc., New York, 2001.
3. R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, John Wiley & Sons, Inc., New York,, 2001.
4. A. K. Jain, M. N. Murty, and P. J. Flynn, "Data Clustering: A Review," *ACM Computing Surveys*, Vol. 31, No. 3, September 1999, pp. 264-323.
5. S. S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*, Artech House, Norwood, MA, 1999.