No Fuzzy Creep!
A Clustering Algorithm for Controlling Arbitrary Node Movement

Francis Harvey
EPFL-IGEO-SIRS
GR-Ecublens
CH-1015 Lausanne
Switzerland
Francis.Harvey@dgr.epfl.ch

François Vauglin
IGN-COGIT Laboratory
2 avenue Pasteur
F-94160 Saint-Mandé
France
Francois.Vauglin@ign.fr

ABSTRACT

A perennial problem in vector overlay is fuzzy creep. Commercial vector overlay algorithms resolve near intersections of lines employing arbitrary node movement to align two chains at nodes selected randomly in the area of an epsilon band. While this solution is effective in reducing the number of sliver polygons, it introduces distortion. In some situations this distortion may be tolerable, but in others it may produce positional errors that are unacceptable for the cartographic or analytical purpose. Our research aims to provide an extension of overlay processing that provides a solution for GIS uses that require more exact control over node movement. The key to this is a robust, non-distorting cluster analysis. The cluster algorithm we present fulfills two goals: 1) it selects nodes based on an nearness heuristic, 2) it allows the user to fix the position of one data set's nodes and moves the other data set's nodes to match these position. In this paper we review existing cluster algorithms from the computational geometry and analytical cartography literature, evaluating their heuristics in terms of the potential to avoid fuzzy creep. Grouping the algorithms into a bit-map and fuzzy-detection types, we discuss the advantages and disadvantages of each approach for controlled near intersection detection. Based on the results of this analysis, we present a algorithm for non-distortive geometric match processing, the basis for our work on geometric match processing.

1. THE PROBLEM WITH FUZZY CREEP

Vector overlay is utilized for a diverse range of purposes to combine geographic information. These purposes place numerous positional accuracy demands that we find are only partially served by existing vector overlay algorithms. All geographic data contains some positional inaccuracy, processing should not increase inaccuracy. We find there is ample need for vector processing algorithms that provide more control.

A crucial problem in current vector overlay algorithms is fuzzy creep (Pullar, 1990; Pullar, 1991; Pullar, 1993; Zhang & Tulip, 1990). Fuzzy creep is

the arbitrary movement of nodes during overlay processing resulting from node snapping, centroid assignment, and induced intersections (Pullar, 1991). This is the result of using a fuzzy tolerance (also known as epsilon tolerance) to resolve near intersections, that otherwise can turn into splinter (or spurious) polygons. Because of its great advantages for resolving near intersections and numerical inaccuracies in processing this type of vector overlay, more commonly known as fuzzy vector overlay (Guevara & Bishop, 1985), is the most common algorithm. Without this algorithm, overlay would be encumbered by a vast amount of spurious polygons, greatly inhibiting the analytical potential of this quintessential GIS operation (Goodchild, 1978).

Still, in spite of great utility, vector overlay algorithms may introduce undesired side effects. These issues are especially pertinent for purposes that require a more exacting control of the overlay operation, especially in terms of positional accuracy. Current applications of fuzzy vector overlay can introduce arbitrary movement of geometric features up to, or even greater, than the epsilon tolerance (Pullar, 1993; White, 1978). The limitation to one fuzzy tolerance for all data sets reduces control possibilities yet further.

We have addressed this broad set of problems in earlier work (1994, 1996) on geometric matching, and in this paper present the continuation of our efforts with a focus on cluster analysis. Briefly, this work has already outlined an algorithm for controlling the movement of nodes by employing multiple tolerances. We distinguish between a match tolerance for the more accurate data set, and a shift tolerance for the less accurate data set. It is possible to align features (without any loss of positional accuracy) from the less accurate data set with features from the more accurate data set.

As the number of digital data sets grows we expect to find an increase of the situation when accurate digital geographic data is combined with less accurate data, i.e. situations when data from field notebooks is combined with digital topographic data, or remote sensing data is combined with precision survey data. A multitude of applications will require functions that combine data, but retain the positional accuracy of the most accurate data set.

Cluster analysis "organizes data by abstracting underlying structure either as a grouping of individuals or as a hierarchy of groups" (Jain & Dubes, 1988). It is crucial to controlling fuzzy creep in vector overlay processing, and has already received attention in analytical cartography. Zhang and Tulip (1990) point to the potential loss in positional accuracy due to fuzzy creep. Pullar (1993), at AutoCarto 11, describes how naively implemented clustering algorithms can lead not only to considerable loss of positional accuracy through fuzzy creep, but even the actual disappearance of features. Beyond his proposal for a technique to control cumulative arbitrary node movement, we present in this paper an algorithm for limiting, and in certain cases, eliminating fuzzy creep.

In the next section we describe existing clustering algorithms used in vector overlay. The following section describes the requirements geometric matching places on clustering algorithms. The conclusion and summary discuss the results, further steps in our research, and possibilities for additional developments.

2. EXISTING CLUSTERING ALGORITHMS

In this section we review existing clustering algorithms for vector overlay. We focus on on each algorithm's maintenance of positional accuracy, computational demands, and suitability for implementation in fuzzy overlay processing.

Approaches using integer and rational arithmetics (Cook, 1978; Franklin, 1987; Wu & Franklin, 1990) have their merits, but computational demands restrict their usefulness for our purposes. Earlier, computer resources were limited and the fuzzy vector overlay using a band-sweep approach became the most successful and common technique. The band-sweep approach means that only a subset of each data set is loaded in memory at a time (White 1978) improving processing efficiency enormously. The reduction in memory demands, the ability to profit from the speed of floating point calculations, and deal with computational imprecision led to its success.

We distinguish between two basic approaches to clustering in computational geometry, bit-map and vector. Here we focus our examination of clustering algorithms on avoidance of fuzzy creep, general computational complexity, and ease of implementation in fuzzy vector overlay processing. First, we will review bit-map approaches.

We identify three bit-map approaches: continuous relaxation, coincidence calculation, and rasterized vector grid. Continuous relaxation is a technique that comes from remote sensing to find correspondences to vector data bases. It aims to construct homogeneous segmented areas based on, a priori probabilities, a proximity relation, a compatibility function, and the definition of an influence function (Lemarié & Raynal, 1996). It is effective for detecting changes in a vector data base, but is very complex and very sensitive to the chosen parameters. It is also very computationally complex and not easily optimized. This approach was developed for raster/vector comparisons, its utility for the cluster analysis of vector data sets remains questionable without further work.

Coincidence calculation actually consists of different techniques. The similarity in these techniques is the basic calculation method. After overlapping polygons (or raster areas) are identified, the similarity parameter is determined using the areas. The similarity parameter gives a probability that the two polygons are the same. A distance parameter can also be calculated. This gives

the relative distance between the areas. The major problem with this approach for our purposes is its limitation to areas.

The rasterized vector grid approach begins with vector data. This data is rasterized into a defined grid and then either the continuous relaxation or coincidence calculation approach is used for cluster analysis.

The restraint to areas in the bit-map approaches is a considerable problem. If the resolution of the raster corresponds to the known accuracy of the data set, these approaches may be quite valuable. However, in cases when the positional accuracy is greater than the cell resolution, rasterizing artificially limits accuracy to the cell size. Fuzzy creep would remain an issue in these cases. In any case, because fuzzy vector overlay processing is not designed around the algorithmic requirements of bit-map cluster analysis, considerable computational inefficiencies could result from implementing these approaches.

Vector approaches to cluster analysis generally allow more exact control over the analysis, but are computationally more complex. Their largest advantage for vector overlay is the ease of integration into existing vector overlay processing algorithms. Vector cluster analysis rests on proximity analysis merging clusters until a condition is met (Jain & Dubes, 1988).

Milenkovic (1989) presents the simplest approach to clustering vector data. His method merely tests if any points are found within an epsilon band, if so they are merged. This operation is repeated until there are no more nodes within the epsilon band. Of course, this leads to a high potential for fuzzy creep.

The approach to clustering in Odyssey's Whirlpool overlay processor, does a somewhat better job. Although the problem of fuzzy creep is recognized, it still allows arbitrary node movement (Chrisman, Dougenik, & White, 1992; White, 1978). Based on this work and others, other approaches were proposed that strive to control fuzzy creep. Zhang and Tulip (1990) specifically address the problem of induced intersections that result from arbitrary node movement. Their approach is based on a proximity matrix that relates objects and the analysis of the matrix uses hierarchical classification. All nodes in the same epsilon band are candidates for merging. Only nodes whose epsilon bands overlap reciprocally are merged. This effectively controls fuzzy creep to the extent of the overlapping epsilon tolerances.

David Pullar proposed an approach similar to Zhang and Tulips (Pullar, 1993). Most notably he describes several constraints that define the clustering. First, a new point must be within the epsilon tolerance of an existing node. Second, to be merged, a node must lie within the epsilon tolerance of the cluster center. Third, in the resulting data set two points cannot share an epsilon tolerance. The constraints are very valuable, but maintaining them in his described implementation is very difficult

These constraints and approaches provide the most substantial base for our development of a clustering algorithm that controls fuzzy creep. Because of its affinity to cartographic data processing, our implementation will build on Pullar's constrained clustering algorithm. In the next section we will look at our requirements for cluster analysis in geometric match processing in detail.

3. GEOMETRIC MATCHING REQUIREMENTS FOR CLUSTERING ALGORITHMS

The requirements for a clustering algorithm that supports geometric matching are few and simple:

1) It eliminates or minimizes fuzzy creep.
2) It gives precedence to the more accurate (match) data set.
3) It selects nodes to merge based on a nearness criteria.
4) The match tolerance may not be greater than the smallest distance between nodes in the match data set.
5) It merges all nodes in each cluster.

In the examples we have thought of, we frequently end with cases that can only be resolved by considering semantic information that at the moment is not available in geometric match processing. This is the most serious caveat for this method and need further work. As geometric match processing stands, it cannot successfully resolve all intersections. The basic limitation is the distance between match and shift nodes. If the shift tolerance is greater than the distance between nodes, they will merge during processing.

Only the judicious application of match and shift tolerances can fulfill the requirements with the clustering algorithm we propose here. To eliminate fuzzy creep, the match tolerance must be set to zero, if it is greater, fuzzy creep is only limited to the value of the match tolerance. Furthermore, the more accurate data set must receive the match tolerance. Following the nearness criteria leads to a trade-off between false positives and complete merging.

The match tolerance must be smaller than the smallest distance between match data set nodes. This is necessary to prevent the creep or collapse of whole clusters. It is also necessary that all nodes in each cluster be merged to assure complete resolution of all possible node merges.

4. CLUSTERING FOR GEOMETRIC MATCHING

Following the band-sweep approach to overlay, we now present a clustering algorithm for use in geometric matching.

Vector overlay using the band-sweep algorithm consists of five steps (White, 1978). Cluster analysis is necessary in several of these steps. Basically, after breaking the chains down into monotonic segments cluster analysis is called for to merge nodes.

Based on Pullar's constrained clustering algorithm (1993), our clustering algorithm distinguishes itself by the priority by which it evaluates match data set nodes. This enhances Pullar's algorithm, and we believe this makes it a must more useful technique. First, if a overlay has to consider $n_m + n_s$ nodes as clustering center points, clustering for geometric matching needs only process match data set nodes. Shift data set nodes are always cluster elements, never cluster centers. Further, because of the requirement that match tolerance be greater than the smallest distance between match nodes, no selection of cluster centers is ever required.

There are several rules (constraints) we have set down to describe clustering behavior and avoid erroneous results.

1) No node can be moved a distance greater than its match or shift tolerance.
2) Nodes with overlapping match tolerances cannot be merged. This would violate the second requirement. If this condition is encountered, processing will terminate and pertinent information provided for the user.
3) When the shift tolerances of two nodes overlap, they may only be merged if the Euclidean distance between them is less than the shift tolerance.
4) Merging of nodes is based on proximity to the cluster center.

The cluster algorithm also considers induced and exact intersections resulting from cluster processing. The band-sweep approach has great benefits for dealing with the large number of intersections that can be created during processing.

Different tolerances leads to different potential clustering situations. Figures 1 and 2 present input and output for four different situations grouped by tolerance value ranges. Figure 1 shows the input situations and figure 2 the output of different clusterings. The two left-side groups depict situations with the shift tolerance set to zero. The upper groups illustrate situations when the match tolerance is set to zero. In the lower groups, the match tolerance is greater than zero, on the right the shift tolerance is greater than zero. In all of the eight situations on the left side of the figures, nothing changes during processing. When the tolerances allow movement, changes occur. The normal case for geometric match processing, when the match data set is more accurate than the shift data set, is illustrated in the lower right group. The specific case, when the geometric matching is used to align elements is illustrated in the upper right group.

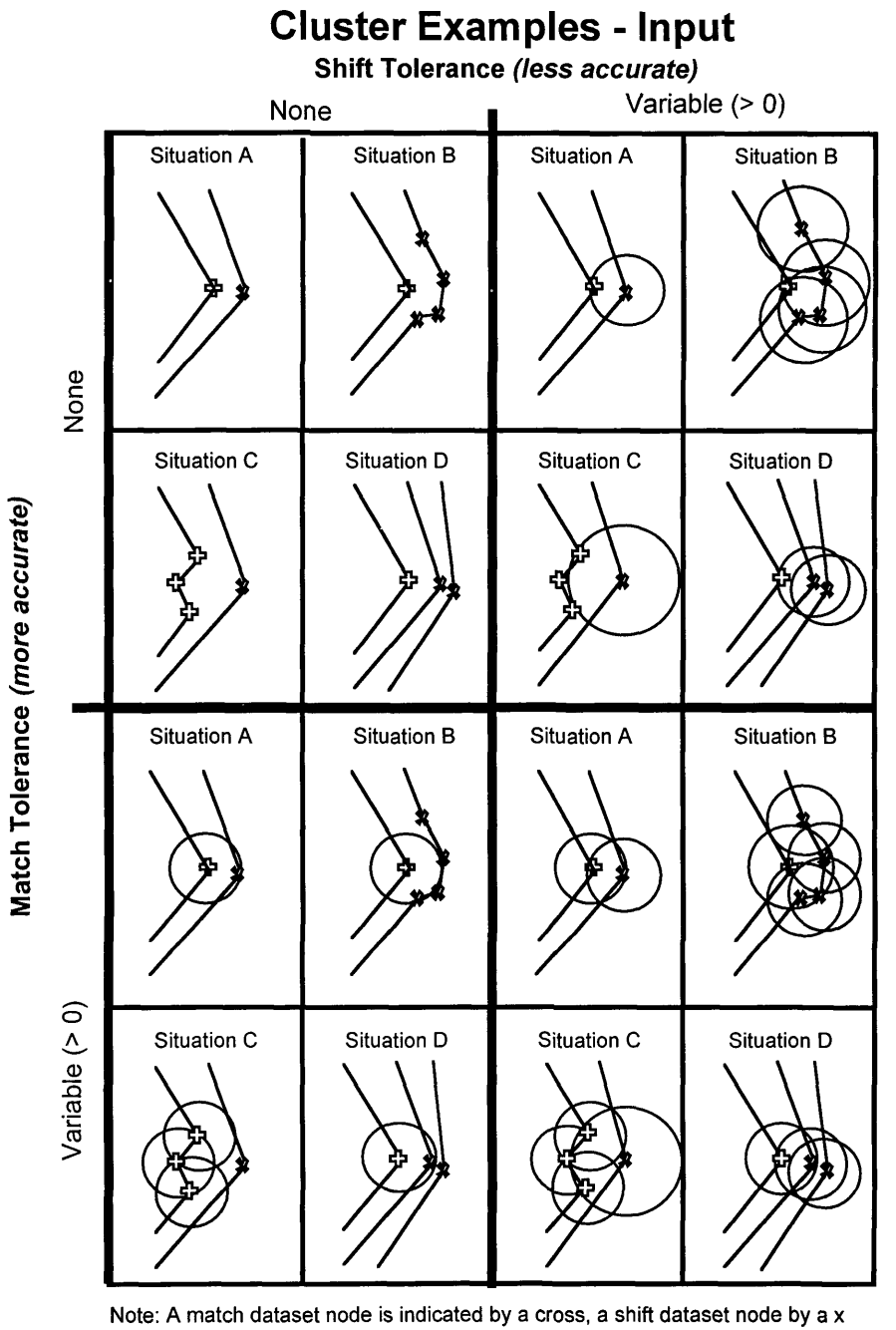

Figure 1 Examples for clusters applying multiple tolerances (Input)

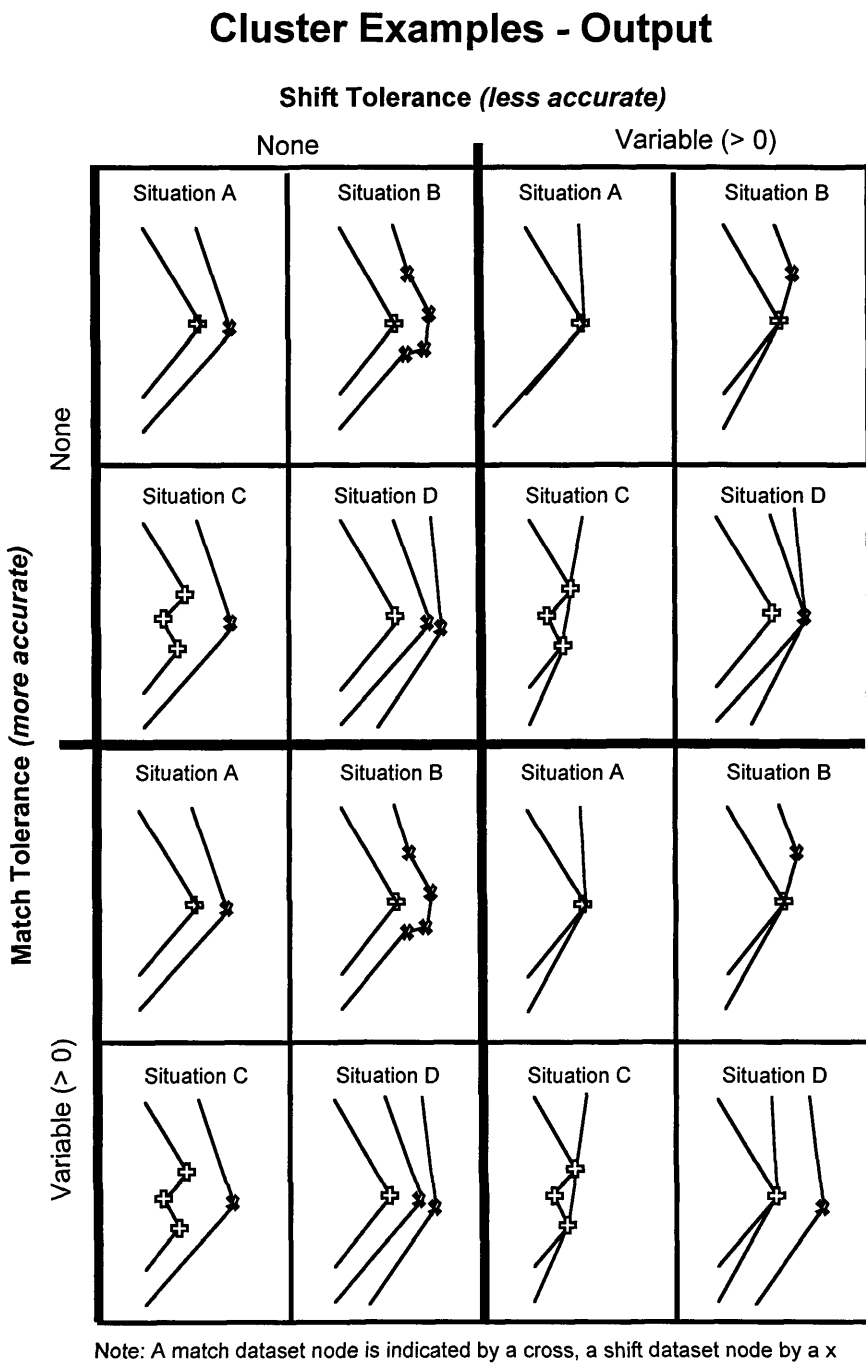

Figure 2 Examples for clusters applying multiple tolerances (Output)

5. SUMMARY AND CONCLUSION

Consideration of fuzzy creep and our implementation of Pullar's algorithm shows that we have only been partially successful in our goal of eliminating fuzzy creep. In summary, Pullar's constrained clustering algorithm reduces possible fuzzy creep to the epsilon value. In geometric matching this is revised to the value of the match tolerance. If the match tolerance is zero, the more accurate data set nodes remain at their locations no fuzzy creep is introduced. As the match tolerance value increases, the amount of fuzzy creep does too. It is impossible to eliminate fuzzy creep if node movement is allowed. In any case, rounding effects and precision limitations will always affect geometric match processing on floating point computers. This issue is only pertinent for extremely accurate work and needs due consideration where the utmost in accuracy is desired.

We conclude the best type of cluster analysis to limit arbitrary node movementfor geometric matching follows vector clustering algorithms. There are several strong reasons for this conclusion:

- it fits within fuzzy overlay processing
- clustering is integral to matching
- extension of existing algorithms
- it can be extended to include feature based parameters

Clearly this approach to clustering in the context of geometric match processing is still limited. We find there is still need to consider statistical methods for determining accuracies and resolving clusters, include feature-based semantics in the geometrification of cluster analysis, preserves shapes, and preserve directions.

At this point we are only able to complete a broad-brush cluster analysis. At least we preserve topology. Based on this work, we believe further improvements will require feature-orientated cluster analysis. Our first thoughts in this direction lead us to consider adding information to the vector data set, for example extending the data structure to include the original x and y locations and the feature epsilon tolerance (x_c, y_c, ε).

References

Chrisman, N. R., Dougenik, J., & White, D. (1992). Lessons for the design of polygon overlay processing from the Odyssey Whirlpool algorithm. In International Symposium on Spatial Data Handling, Proceedings, . Charleston, NC: SDH.

Cook, B. G. (1978). The Structural and Algorithmic Basis of a Geographic Data Base. In G. Dutton (Eds.), Harvard Papers on GIS, First International

Advanced Study Symposium on Topological Data Structures for Geographical Information Systems Cambridge: Harvard University.

Franklin, W. R. (1987). A polygon overlay system in prolog. In AutoCarto 8, Proceedings, Vol. 1 (pp. 97-106). Baltimore, MD: ACSM.

Goodchild, M. F. (1978). Statistical Aspects of the Polygon Overlay Problem. In Harvard Papers on GIS, First International Advanced Study Symposium on Topological Data Structures for Geographical Information Systems Cambridge: Harvard University.

Guevara, J. A., & Bishop, D. (1985). A fuzzy and heuristic approach to segment intersection detection and reporting. In AutoCarto 7, Proceedings, 1 (pp. 228). Washington D.C.

Harvey, F. (1994). Defining unmoveable nodes/segments as part of vector overlay. In T. C. Waugh & R. G. Healey (Ed.), Sixth International Symposium on Spatial Data Handling, 1 (pp. 159-176). Edinburgh, Scottland: T. C. Waugh IGU Commision on GIS/Association for Geographic Information.

Harvey, F., & Vauglin, F. (1996). Geometric match processing: Applying Multiple Tolerances. In M. J. Krakk & M. Molenaar (Ed.), The Seventh International Symposium on Spatial Data Handling (SDH'96), Proceedings, Vol. 1 (pp. 4A-13 - 4A-29). Delft, Holland: International Geographical Union (IGU).

Jain, A. K., & Dubes, R. C. (1988). Algorithms for Clustering Data. Englewood Cliffs, NJ: Prentice Hall.

Lemarié, C., & Raynal, L. (1996). Geographic data matching: First investigations for a generic tool. In GIS/LIS '96, Proceedings, 1 (pp. 405-420). Denver, Co: ASPRS/AAG/URISA/AM-FM.

Milenkovic, V. J. (1989). Verifiable implementations of geometric algorithms using finite precision arithmetic. In D. Kapur & J. L. Mundy (Eds.), Geometric Reasoning (pp. 377-401). Cambridge, MA: MIT Press.

Pullar, D. (1990). Comparative study of algorithms for reporting geometrical intersections. In K. Brassel & H. Kishimoto (Ed.), Fourth International Symposium on Spatial Data Handling (SDH), Proceedings, Vol. 1 (pp. 66-76). Zürich: Waugh, T. IGU/AGI.

Pullar, D. (1991). Spatial overlay with inexact numerical data. In AutoCarto 10, Proceedings, 1 (pp. 313-329). Baltimore, MD: ACSM.

Pullar, D. (1993). Consequences of using a tolerance paradigm in spatial overlay. In R. McMaster (Ed.), AutoCarto 11, Proceedings, 1 (pp. 288-296). Minneapolis, Minnesota.

White, D. (1978). A Design for Polygon Overlay. In Harvard Papers on GIS, First International Advanced Study Symposium on Topological Data Structures for Geographical Information Systems Harvard University.

Wu, P. Y. F., & Franklin, R. W. (1990). A logic programming approach to cartographic map overlay. Computational Intelligence, 6(2, May 1990), 61-70.

Zhang, G., & Tulip, J. (1990). An algorithm for the avoidance of sliver polygons and clusters of points in spatial overlay. In Fourth International Conference on Spatial Data Handling (SDH), Proceedings, (pp. 141-150). Zurich: