CONSEQUENCES OF USING A TOLERANCE PARADIGM IN SPATIAL OVERLAY

David Pullar
Environmental Systems Research Institute
380 New York Street
Redlands. CA 92373
dpullar@esri.com

ABSTRACT

Geometrical algorithms for spatial overlay incorporate a fuzzy tolerance to remove spurious polygons. This fuzzy tolerance may be related to numerical limits for calculating intersections, or related to the variation in the geometry of objects they represent. When the distance between two objects is below the tolerance they are classified as coincident and moved to the same location. Yet using a tolerance as a heuristic to handle decisions of approximate coincidence has some undesirable consequences. Some of the problems encountered are that objects creep outside their allowable tolerance, or the objects geometry is corrupted so that proper topological relations cannot be reconstructed. This paper examines the flaws with applying a tolerance paradigm in spatial overlay, and describes the conditions needed to safely evaluate point coincidence.

INTRODUCTION

Spatial overlay is an analytical tool used to integrate multiple thematic layers into a single composite layer. This involves intersecting all the geometrical objects from each layer and filtering the geometry to remove spurious polygons in the output. This geometrical filtering process relates to techniques for automated scale changing called *epsilon filtering* [Chrisman 1983]. The two geometrical criteria it imposes on the output are; i) creep - no point is moved more than epsilon, and ii) shrouding - no points are left within a distance epsilon. The epsilon distance in map overlay is the geometrical tolerance.

The basis of the filtering process is to resolve point coincidences. That is, if two points are found to be within the geometrical tolerance to one another they are merged and identified as a single point. Most commercial GIS's use a single tolerance as an input parameter to the overlay program for classifying incidence and coincidence relations between objects. Yet there is a need to distinguish between objects that have a well defined geometry with a small tolerance, and objects that have an imprecise geometry. That is we need to distinguish between multiple cases in the same layer where; i) two points are distinct objects separated by a small distance, and ii) two points are meant to represent the same object when they are a small distance apart. This is the rationalization for multiple tolerances.

Most algorithms employ some *tolerance paradigm* to detect coincident points. The tolerance paradigm is a heuristic that says if two quantities are near enough in value then treat them as the same. A simple example will demonstrate the problem with this reasoning paradigm for geometrical algorithms. Figure 1 shows a set of points and their tolerance environments. Each point, called an *epsilon point*, is given by a tuple (x,y,ε) representing an x,y-coordinate and a tolerance radius ε. We arbitrarily chose one epsilon point and begin

testing for point coincident based on the tolerance regions overlapping. When overlapping epsilon points are found they are shifted to coincide exactly. If the points are tested and moved in the sequence shown in figure 1 then it is apparent points will creep from their desired location. One also could imagine the epsilon points as connecting line segments, and the segments degenerating to a point.

The flaw in our reasoning for the tolerance paradigm is in the transitivity of point coincidence. That is, we cannot naively assume that if ($P_1 = P_2$) and ($P_2 = P_3$) implies ($P_1 = P_3$) when the test used to evaluate equality is approximate.

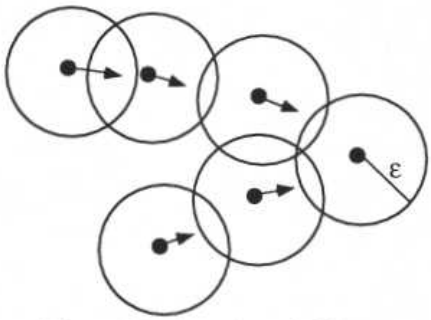


Figure 1. Testing point coincidence

Applying the tolerance paradigm naively causes algorithms to fail or give an incorrect result. And even if applied judiciously the tolerance paradigm has some severe consequences. Recent research in the field of solid modelling has discussed this topic at length. Robust algorithms for intersecting polyhedral objects are known, but it is admitted in a worst case scenario the algorithms will allow objects to collapse to a point [Milenkovic 1989]. This degenerate behavior for computing intersections has even worse consequences when the tolerances become larger and there are multiple polygons intersected. The cause of the problem is in the way points are tested for approximate coincidence before moving them.

CONDITIONS FOR POINT COINCIDENCE

The conditions for resolving coincidences between epsilon points with multiple tolerances is similar to the conditions stated for epsilon filtering. Namely, two geometrical criteria are imposed on the output; i) creep - no point is moved more than epsilon, and ii) shrouding - no points are left within a distance epsilon. The question arises as to the conditions to be fulfilled when dealing with multiple tolerances?

We can define the first criteria creep with respect to multiple tolerances in a straight forward way. A point cannot be moved a distance greater than its epsilon tolerance. What is not straight forward is how to apply the shrouding criteria and make sure points are separated by some minimum distance. This raises two questions when dealing with multiple tolerances;

1. What tolerance will be used to compare two epsilon points?
2. How is the tolerance updated when merging epsilon points?

One obvious way to deal with the first question is to say points are coincident if their tolerance regions overlap. Another way is to say epsilon points maintain a minimum

separation distance. We explore both possibilities.

Overlapping Tolerance Regions

Lets assume the shrouding criteria is based on a geometrical condition, namely the tolerance regions for two points must not overlap. This answers the first question by stipulating that if the sum of the radii for the epsilon regions is greater than the distance between their point centers then they need to be merged. Figure 2 shows this situation.

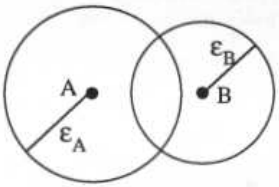


Figure 2. Tolerance environments for two points overlap

To answer the second question we need a way to update the tolerances for two epsilon points. Some different scenarios for updating the tolerance ε_{AB} after merging two points 'A' and 'B' are;

1. ε_{AB} = maximum(ε_A,ε_B), i.e. the maximum tolerance.
2. ε_{AB} = minimum(ε_A,ε_B), i.e. the minimum tolerance.
3. $\varepsilon_{AB} = 2/[(\varepsilon_A)\text{-}1+(\varepsilon_B)\text{-}1]$, i.e. the weighted sum of the two tolerances.
4. $\varepsilon_{AB} = \varepsilon_A \cup \varepsilon_B$, i.e. the smallest enclosing sphere within their union.
5. $\varepsilon_{AB} = \varepsilon_A \cap \varepsilon_B$, i.e. the smallest enclosing sphere within their intersection.

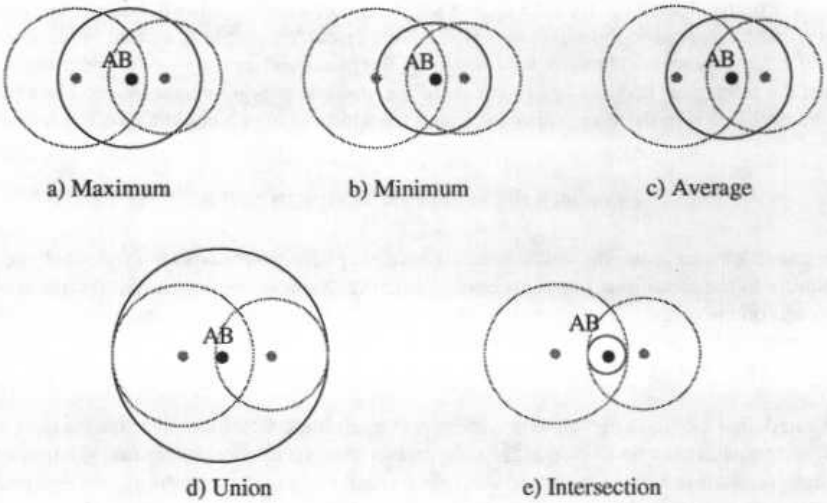
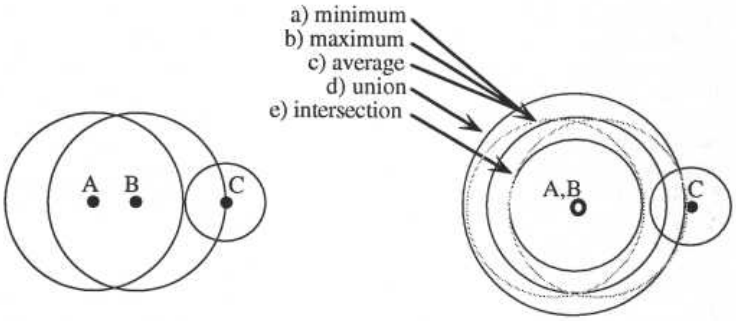


Figure 3. Updating the tolerance regions for point coincidence

Figure 3 shows each of the above methods. By examining some specific point configurations we can easily show that none of the methods are adequate. For instance, consider the three epsilon points 'A', 'B', and 'C' and their associated tolerance environments in figure 4a. Points 'A' and 'B' are found to be close and are merged to form

'AB'. The first four methods of updating the tolerances for 'A' and 'B' - e.g. maximum, minimum, average, and union - all cause overlap with the tolerance region for 'C'. This is illustrated in figure 4b. But if 'C' is merged with 'AB' this will contradict the creep criteria. That is the new point will likely be outside the tolerance region for 'C'. The fifth method, e.g. intersection, is also inappropriate because when the tolerance regions for two points are just overlapping then the updated tolerance will converge to zero. This would easily lead to instability in determining the coincidence between points.



a) Comparing points 'A','B', and 'C' b) Point 'A' and 'B' are merged

Figure 4. Coincidence Test

Therefore our first approach for defining the shrouding criteria has some basic flaws. None the less, some researchers in the field of computational geometry have used approaches similar to this in their work. Segal [1990] uses the union of tolerance regions to analyse point coincidence, but does so at the expense of relaxing the creep criteria. Fang and Bruderlin [1991] use a combination of computing both the union and intersection of the tolerance regions for detecting ambiguities when analysing point coincidences. The algorithm is re-run when an ambiguity is discovered using different estimates for the tolerances.

Minimum Separation

From simple reasoning we have shown the flaw in merging points with overlapping tolerance regions to enforce the shrouding criteria. An alternative to is base the shrouding criteria on maintaining a threshold separation between two points. The separation distance must be related to the tolerances of the points. An obvious possibility is to use any one of the first three criteria for updating clusters discussed in the last section, namely the minimum, maximum or average tolerance for the points. Therefore an alternative test for shrouding is that the separation d_{AB} between two points 'A' and 'B' be less than;

1. $d_{AB} < \text{maximum}(\varepsilon_A,\varepsilon_B)$, i.e. the maximum of the two tolerances.
2. $d_{AB} < \text{minimum}(\varepsilon_A,\varepsilon_B)$, i.e. the minimum of the two tolerances.
3. $d_{AB} < 2/[(\varepsilon_A)\text{-}1+(\varepsilon_B)\text{-}1]$, i.e. the weighted average of the two tolerances.

It is natural to assume the updated tolerance for an output point is also computed from the respective maximum, minimum, or average of the tolerances. One could easily expect the first possibility of using the maximum of the two tolerances is violated by inspection of figure 4. Less obvious is the fact that using a weighted average as a shrouding criteria and then updating the tolerances also will violate the creep criteria for certain input data. We do

not prove this, but have found this to be the case from running several tests with randomly generated data sets. Rather, we will set out to prove that a lower bound using the minimum tolerance as a separation criteria may be achieved. From empirical testing we could find no test case that violated this condition when merging epsilon points in the most compact way.

Thus, this paper proposes that coincidence of epsilon points may be solved in a consistent way to satisfy the following geometrical criteria;

1. An input point cannot be moved more than its epsilon tolerance to merge with an an output point, i.e. if 'A' is moved to 'B' then the distance $d(A,B) < \varepsilon_A$.
2. Epsilon points in the output set are separated by a minimum distance, this lower bound is determined by pairwise comparisons to be the minimum epsilon tolerance, i.e. output epsilon points 'A' and 'B' are separated by at least the distance $\text{minimum}(\varepsilon_A,\varepsilon_B)$.
3. When two epsilon points are merged a new point center is located at their mean position, and a new tolerance is computed as the minimum of the two epsilon tolerances, i.e. if epsilon points 'A' and 'B' are merged then $\varepsilon_{AB}=\text{minimum}(\varepsilon_A,\varepsilon_B)$.

We have found that these are the only feasible conditions that may be met for unambiguously solving point coincidence. To show this, we first re-state the point coincidence problem in a more formal way. We use concepts from graph theory to solve for point coincidence as a graph location problem. We then prove that the three conditions stated above are satisfied.

A MODEL FOR POINT COINCIDENCE

To prove the minimum separation criteria is valid for all inputs we need to define the properties of output points that satisfy the point coincidence relations. The best way to describe this problem, and its solution, is as a point clustering problem. Hartigan [1975] describes clustering as the 'grouping of similar objects'. In our case the objects are points with their associated tolerance, and they are grouped to new centers that satisfy the coincidence relations. The clustering is also chosen to maximize the separation criteria by stipulating that point coincidences minimize some measure of dissimilarity. Hartigan describes several dissimilarity measures, one popular method is to group the elements in a way that minimizes the spread of points in each group. Minimizing the spread is interpreted as minimizing the sum of the squared lengths from cluster points to their center. This is called a *sum-of-squared-error clustering* [Duda and Hart 1973].

The P-median Location Problem

The sum-of-squared-error clustering is closely related to finding the medians of a graph, this is called the *p-median* problem [Christofides 1975]. The Euclidean p-median problem is described in the following way. Given a set $X=\{p_1, p_2, ..., p_n\}$ of n points (x,y), find a set X' of m points $\{p'_1, p'_2, ..., p'_m\}$ so as to *minimize* the expression;

$$\sum_{i=1}^{n} \min_{1 \le r \le m} \{ \| p'_r - p_i \| \} \qquad (1)$$

where ||..|| designates the metric distance between point centers.

Intuitively, we wish to minimize the sum of the radii that enclose points of X by circles located at centers of X'. We also refer to the points of X' as cluster centers.

The p-median problem has some nice set-theoretic implications. The set of points from X associated with a cluster center define a *set-covering* of X. The points of X are grouped into sets $X_1,...,X_m$ according to way the circles located at cluster centers of X´ cover points in X. These are called the *covering sets*, such that;

$$\bigcup_{r=1}^{m} X_r = X \qquad (2)$$

In addition the sets $X_1,...,X_m$ are pair-wise disjoint. This is called a *set-partitioning* of X, such that;

$$X_i \cap X_j = \varnothing, \quad \forall i,j \in \{1,...,m\} \qquad (3)$$

These set-theoretic properties are used to define point coincidences in a consistent way. We will adapt the p-median problem to deal with epsilon points. We now re-state the problem, and call it a *distance constrained* p-median problem.

Constrained Clustering

The distance constrained Euclidean p-median problem is described in the following way. Given a set $X=\{p_1, p_2,..., p_n\}$ of *n* epsilon points (x,y,ε), find a set X´ of epsilon points $\{p'_1, p'_2,..., p'_m\}$ where $m \leq n$ so as to *minimize* the expression;

$$\sum_{i=1}^{n} \min_{1 \leq r \leq m} \{ \| p'_r - p_i \| \}, \quad \| p'_r - p_i \| < \varepsilon_i \qquad (4a)$$

and

$$\| p'_r - p'_s \| < \text{minimum } (\varepsilon'_r, \varepsilon'_s), \quad \forall r,s \in \{1,...,m\} \qquad (4b)$$

where ||..|| designates the metric distance between epsilon point centers.

As in (1) we are minimizing the sum of the radii that enclose points of X by circles located at cluster centers of X´. But equation (4a) requires the distance between a cluster center p'_r and an input point p_i is constrained to be less than the epsilon tolerance ε_i for that point. Equation (4b) additionally stipulates that a minimum separation is maintained between clusters centers.

Notice that a variable number of cluster centers $m \leq n$ is permitted. The problem now resembles a clustering procedure rather than a graph location problem, for this reason we refer to the solution of the distance constrained Euclidean p-median problem simply as a *constrained clustering*. The lower limit to the number of cluster points *m* is determined by the minimum number of points that will define a set-covering of X based upon equation (4b). The number of clusters is given by the solution set that satisfies both conditions.

The constrained clustering defines a set-partitioning of X. Each epsilon point of X is clustered to the nearest cluster point from the set X´. So any two points p_i,p_j clustered together are considered part of the same equivalence class based on the relation p_i *is-coincident-to* p_j. Since coincidence is an equivalence relation then by definition the relation is reflexive, symmetric and transitive. This property is used to avoid the inconsistencies

found in naive algorithms for comparing points.

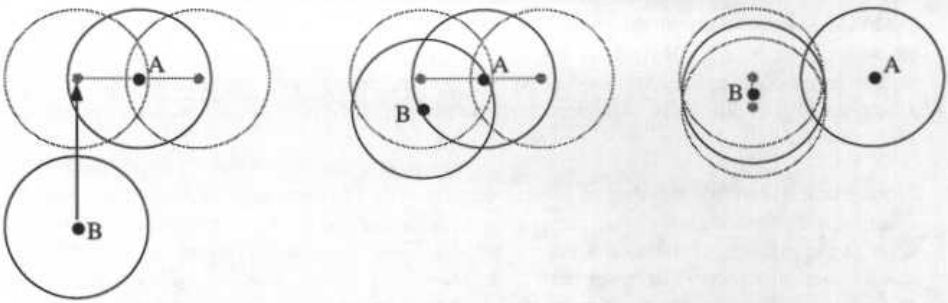
Proof of Constrained Clustering Conditions

We need to prove a constrained clustering fulfills the three geometrical conditions stated in the last section. These conditions will be defined in terms of a clustering procedure to decide point coincidence relations on a set of epsilon points.

Definition. Given a set of epsilon points $X=\{p_1, p_2,..., p_n\}$ we partition X into subsets $X_1,...,X_m$. Each subset X_k defines a cluster set with a representative point p'_k chosen as the cluster center, the set of *m* cluster centers is denoted as the set X'. The relation between the epsilon points in the sets X and X' is called a constrained clustering when the following conditions are satisfied;

1. $\| p'_r - p_i \| < \varepsilon_i$ (from eq. 4a)
2. $\sum_{i=1}^{n} \min_{1 \le r \le m} \{ \| p'_r - p_i \| \}, \quad \| p'_r - p_i \| < \varepsilon_i$ (from eq. 4a)
3. $\| p'_r - p'_s \| <$ minimum $(\varepsilon'_r, \varepsilon'_s), \quad \forall r,s \in \{1,...,m\}$ (from eq. 4b)

Proof. Lets assume a constrained cluster exists satisfying conditions 1 and 2. For now lets disregard condition 3, and allow clusters to be separated by less than the minimum tolerance. Without loss of generality, assume there are two cluster centers $p'_r, p'_s \in X'$ which are separated by a distance less than minimum$(\varepsilon'_r, \varepsilon'_s)$. There must also exist extreme[1] points belonging to the cluster sets $p_i \in X_r$, $p_j \in X_s$ which lie within the distance minimum$(\varepsilon'_r, \varepsilon'_s)$ to one another. Since these points are within the minimum tolerance to one another they are free to group together without violating condition 1, and form a new cluster with a smaller diameter. Therefore, there must be another way of clustering the points which gives a smaller sum of lengths between cluster centers and points from X. This violates condition 2, and to avoid the contradiction we conclude condition 3 must be true.



a) Move 'B' closer to 'A' b) 'A' and 'B' within minimum tolerance c) Re-clustering to satisfy minimum diameter clusters

Figure 5. Demonstrates proof of constrained clustering

As an example we can examine the arrangement of points shown in figure 5. Diagram a) shows two clusters that satisfy conditions 1-3. We examine the effect of moving the two

[1]Extreme points are the points on the convex hull of a set of points [Preparata and Shamos 1985]

clusters closer together, until in diagram b) they are separated by less than their minimum tolerances. It is evident that the points could be re-clustered to obtain a more compact cluster with a smaller sum of lengths between cluster centers and input points, as shown in diagram c).

This means we can place an upper bound on the distance points can be moved, and a lower bound on the separation between final point centers. Even more importantly, we have defined the coincidence relation based upon the constrained clustering. Points that belong to the same cluster $\{p_i,p_j \in X_r\}$ are coincident, otherwise they are non-coincident. By using this set membership relation to test for point coincidence we satisfy the transitivity rules and avoid inconsistencies.

CONCLUSION

This paper has examined how a tolerance paradigm is used in geometrical algorithms. Each point is assigned a tolerance and is referred to as an epsilon point. The tolerance paradigm determines what epsilon points are coincident, and subsequently merges them. In determining coincidences certain geometrical properties are desired; these are i) creep to prevent input points from drifting too far, and ii) shrouding to maintain some separation between points. The paper focuses on a way to test for coincidence of epsilon points with multiple epsilon tolerances, and several solutions are discussed. We found one solution that uses a clustering approach to merge epsilon points is a suitable model for point coincidence.

The problem of clustering epsilon points, which we named constrained clustering, is described using set-theoretic principles from graph location theory. We describe a variation of the Euclidean p-median problem which constrains the distances between points to satisfy the two geometrical criteria for creep and shrouding. Using this approach, point coincidence is defined as an equivalence relation over a set of epsilon points. This allows us to make set-theoretic conclusions about epsilon points. For instance, we can now unambiguously say that if (P_1 is-coincident-to P_2) and (P_2 is-coincident-to P_3) implies (P_1 is-coincident-to P_3).

Being able to cluster epsilon points and guarantee geometrical properties has beneficial consequences for designing a multi-tolerance overlay algorithm. It provides verification conditions that is used in a correctness proof for the map overlay algorithm [Pullar 1991].

Another implication of these results is that to obtain upper and lower bounds for creep and shrouding we must solve a geometric location problem. The Euclidean p-median problem has a complexity classed as NP-hard, and even an approximate solution requires an efficient clustering algorithm [Megiddo and Supowit 1984].

REFERENCES

Chrisman N., 1983, Epsilon Filtering: A Technique for Automated Scale Change. *Proceedings 43rd Annual Meeting of ACSM*: p.322-331

Christofides N., 1975, *Graph Theory: An Algorithmic Approach.* Academic Press.

Duda R., and Hart P., 1973, *Pattern Classification and Scene Analysis.* Wiley Interscience.

Fang S. and Bruderlin B., 1991, Robustness in Geometric Modeling - Tolerance-Based Methods. *Proceeding International Workshop on Computational Geometry CG'91*, Switzerland, Lecture Notes in Computer Science, #553, Springer-Verlag, Editors H.Beeri and H.Noltemeier. p.85-102

Hartigan J.A., 1975, *Clustering Algorithms*. Wiley, New York.

Megiddo N., and Supowit K., 1984, On The Complexity Of Some Common Geometric Location Problems. *SIAM Journal of Computing* 13(1): p.182-196

Milenkovic V.J., 1989, Verifiable Implementations Of Geometric Algorithms Using Finite Precision Arithmetic. In: *Geometrical Reasoning*, editors D. Kapur and J. Mundy, MIT Press, Pennsylvania.

Preparata F.P. and Shamos M.I., 1985, *Computational Geometry*. Springer-Verlag, New York.

Pullar D.V., 1991, Spatial Overlay with Inexact Numerical Data, *Proceedings Auto-Carto 10*, Baltimore. p.313-329

Segal M., 1990, Using Tolerances to Guarantee Valid Polyhedral Modeling Results. *Proceedings SIGRAPH*: p.105-114