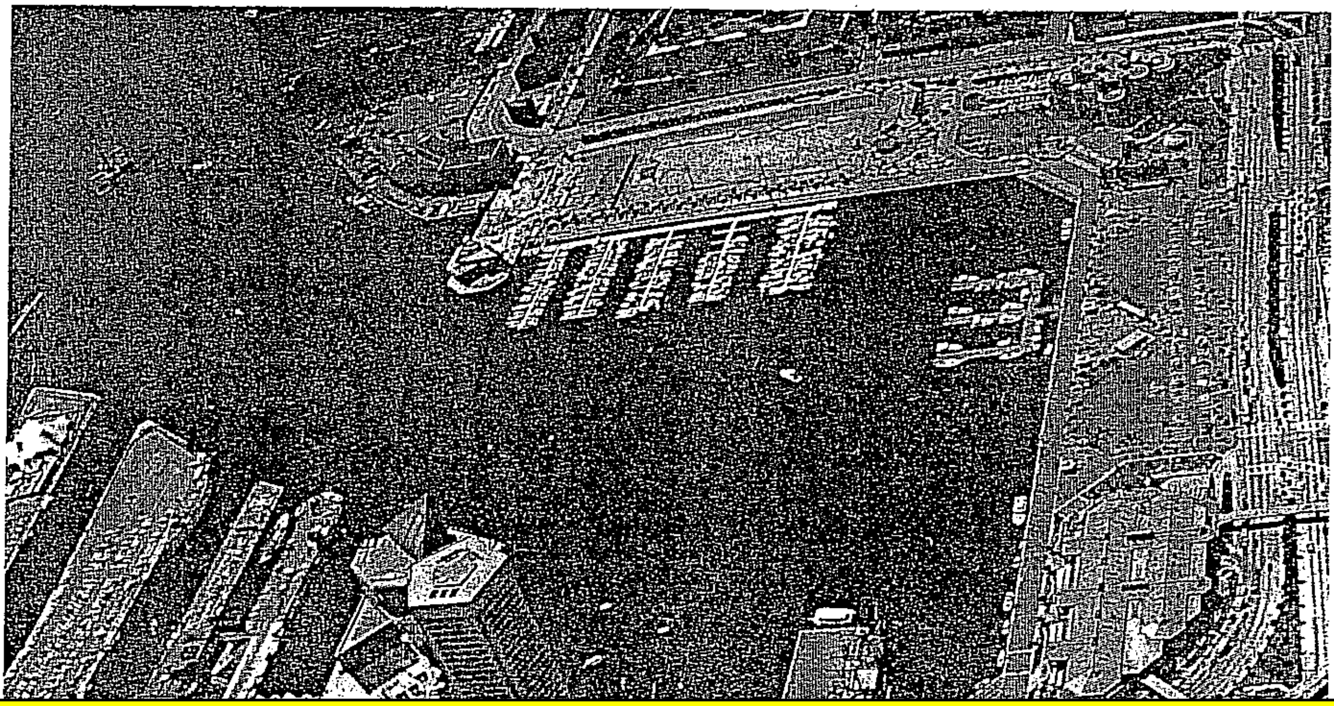
AUTO CARTO 8

Proceedings

Eighth International Symposium on Computer-Assisted Cartography

Baltimore, Maryland
March 29 — April 3



Frank, A.U. "Overlay Processing in Spatial Information Systems." Paper presented at the Auto-Carto 8, Baltimore, Maryland (March 29-April 3, 1987), March 29-Aprli 3 1987.

OVERLAY PROCESSING IN SPATIAL INFORMATION SYSTEMS

Andrew U. Frank
University of Maine
Department Of Civil Engineering
Surveying Engineering
Orono, ME 04469
FRANK@MECAN1 (Bitnet)

1 ABSTRACT

Combining the information in two or more thematic layers - the overlay operation - is a major problem in geographic information systems.

First a framework for information processing during an overlay operation is given. The geometric intersection operation is separated from the treatment of property or attribute values. In the same way that spatial objects are described by geometric and non-geometric properties, a thematic layer is defined as a geometric partition, dividing space into areas, and the property values associated with each area. To overlay two layers, therefore, requires the computation of the intersection of the two geometric partitions and the combination of the property values.

The calculation of intersections of spatial subdivisions are difficult to execute on computers. The designer of a program to do so must cope with the limited precision of computations on computers. The programs presently available exhibit, at least for some special cases, incorrect behaviour and are computationally demanding.

Two routes seem possible for a spatial information system: either the intersection of two geometric partitions is calculated when it is necessary to respond to a question asked by a user, or all possible intersections are calculated when data is integrated into the spatial information system. The first method seems to have advantages for situations where many existing data sets are combined in a one-time effort to produce a new map. The second appears more suitable for situations where a data collection is built for long-term storage and is intended to be continuously updated.

Separating the geometric operations of intersections from the combinations of the attribute data permits the computation of arbitrary attribute combinations and the classification of attribute values without complex geometric computations.

Work on these concepts was partially supported by a grant from National Science Foundation (grant no. Ist- 8609123) and a joint statistical agreement with the U.S. Bureau of the Census. Contributions from Digital Equipment Corp. are also acknowledged. The Swiss Federal Institute of Technology (ETH), Zurich, and Prof. R. Conzett and his group must be thanked for their hospitality while this paper was prepared. The ideas and opinions expressed here, however, are solely those of the author.

2 INTRODUCTION

A Geographic or Land Information System (GIS and LIS, respectively) or, more generally, any modern spatial information system, contains information related to land. Because of their increased power they are rapidly replacing conventional maps as primary tools for spatial analysis. Overlay techniques, well known from manipulations with conventional maps, are of great importance in many different application areas. [Chrisman 1978]

There exists a danger that we may design tools into the GIS which blindly imitate manual operations. Manual operations usually have some limitations which protect users from the most obvious forms of misuse and abuse. Computer operations, however, are fast enough and have fewer other limitations so that naive users easily produce results with little relevance, and often with completely misleading data.

We present here a theoretical analysis of data processing in an overlay operation. The main result is the separation of geometric operations on partitions and the non-geometric combination of data values in layers for thematic specific property values. An improved understanding is helpful, not only for the design and implementation of lower levels of the GIS, but, more importantly, for the design of the user interface. This leads to systems which are easier to learn and use.

Currently a trend towards database oriented design of information systems can be observed. An information system must be based on a database if users expect timely answers and want to be able to update the system as they learn about changes. Such spatial information systems can form focal points for the organization of land related information in public administration. Organization of data stored for long term usage in a database should follow normalization rules. The computation of geometric intersections when data is entered into these systems fits well into this scheme.

3 SPATIALLY RELATED INFORMATION

In this section we will explore some basic properties of spatially related information. Our position here is only conceptual and no assumptions about an implementation are made. Indeed, this paper strives to separate the theoretical issues from implementation details which, all too often, have made discussions difficult to follow and the results not formulated generally enough for application in other contexts.

Information related to a spatial object either describes the geometry of the object or its spatial or other properties. This separation seems to be trivial, but is of fundamental importance.
spatial object = geometric properties + non-geometric properties

This paper deals only with discrete delimited objects with specific property values. It excludes cases where information is thought of as being related to points in continuous space formulated as a function f(x) (where x is a coordinate tuple describing a point) as, for example, with digital terrain models, or magnetic declination, etc.

Generally, spatial objects may be points, lines or areas, but most overlay processing is carried out on areal objects. We will discuss only these, once again because of limitations in space.

3.1 Non-geometric Properties

Non-geometric information is abstractly a pair consisting of property name and value. The name indicates which property (e.g. land use, land value or height above sea level) is described by the value (e.g. "residential", "$30 per sq. feet" or "350 feet"). It is important to realize that the value alone is not sufficient. The property must also contain an indication of how to interpret the value. A property name can be seen as a function which maps from an object to a value for the named property [Shipman 1981].

A property for which each object has a unique value is called an identifying property and the respective value an identifier or key. Common examples are names, social security numbers, etc.

The values for a property are selected from a domain, e.g. the integer numbers, real numbers, or the names of classes of things. For different properties different encodings are appropriate. Stevens proposed four types: ratio, interval, ordinal and nominal [Stevens 1946]. The selection of an appropriate operation to apply to a property depends on which type of measurement is involved. For instance, it is quite obvious that the calculation of an average is not meaningful for nominal data.

With certain properties we will encounter the need for a null value to represent the absence of a value, either because we do not know it or because it is not applicable. This is not included in the customary algebra (e.g. that which is available on real numbers, etc.) and an extension is necessary. Treatment of null values during operations has been discussed in database literature [Date 1982] [Codd 1986], but a simple solution is not yet known. The widespread usage of 0, -1 or 99 to encode "unknown" can mislead the unwary user and these values seldomly integrate well with operations on the properties.

A systematic study using the new theory of abstract data types or multi-valued algebras [Guttag 1977,1978,1980] [Parnas 1972] to model the categories of encoding measurements would be beneficial. It will show which operations are available on which measurement type and could include the propagation of errors and imprecision. It must specify how "unknown" and "not applicable" are to be treated.

3.2 Identifiers As Locators

The combination of geometric and non-geometric properties need not be simple and direct as assumed above, but can be mediated by describing an object with non-geometric properties and, in lieu of a geometric description, by using a reference which identifies another, geometrically described, object. Such references must be property values which select the designated object uniquely (i.e. an identifier or key value). They can be called locators [Frank 1984]. Most often we use a street address, a parcel identifier, or the name of a town as a locator to describe the spatial object to which some attribute data relate.

Spatial object = identifier for other spatial object (locator) + non-geometric properties

To process the data spatially, we can replace the identifier with the geometric properties of the referenced object and derive a spatial object with an explicit geometric description. It is worth noting that locators can be used in a nested fashion; thus an object can be located using an identifier which references an object which in turn uses a locator to reference another geometric object.

The operation of replacing locators by explicit geometric descriptions can be modelled as a join (exactly equi-join) in a relational calculus [Date 1983] [Ullman 1980]. Please note that no geometric processing is necessary for this operation.

Example:

Relation PARCEL consists of geometric description, parcel-id, valuation. Another relation OWNER consists of parcel-id, owner-name. Assuming that parcel-id is a locator, we can use a join to deduce a combination relation PARCEL-OWNER with geometric description and owner-name.

PARCEL

Id	Value	Descr.
64a-Q	$1025.00	N.E. corner 5th & Elm St.
55e-T	$ 900.50	234 Main St.

OWNER

Id	Owner name
81k-N	Margo Foont
64a_Q	Pelman Twilly
55e-T	ACME Trivet Inc.

Join on Id to produce:
(with appropriate projections on attributes)

PARCEL-OWNER

Id	Owner name	Descr
64a_Q	Pelman Twilly	N.E. corner 5th & Elm St.
55e-T	ACME Trivet Inc.	234 Main St.

3.3 Geometric Descriptions Of Objects

The objects in a spatial information system refer to some objects in the real world for which we know the location and extent in space. The determination of geometric properties of objects is always limited to some approximation and generalization due to limited precision in measurements, and also to limited resolution in the representation, e.g. in a computer system. These approximations pose some special problems which are presently poorly understood and need special attention.

The geometric description of objects may be either points (0-dimensional), lines (1-dimensional), areas

(2-dimensional) or volumes (3-dimensional) - for the context of GIS and LIS we usually exclude the third dimension for information retrieval operations, but modelling using volumes is of great interest for geological and geotechnical applications, including ground water flow modelling, etc. [Carlson 1987].

In this paper we will concentrate on a dimension independent treatment, i.e. results should be valid independent from the number of dimensions used [Giblin 1977] [Frank & Kuhn 1986].

Two basic methods for geometric descriptions are used:

1. vector based, where point positions are fixed with coordinate values and objects are described using lines lines running between these points [Corbett 1975]
2. raster based, where space is divided into a (usually regular) grid and object geometry is recorded as a list of grid cells which approximate the object geometry [Samet 1984]

4 GEOMETRIC CONSIDERATIONS FOR OVERLAY OPERATIONS

In this section we will present some theoretical background for the geometric aspects of the the overlay operation which are independent of the descriptive or attribute data associated with the spatial objects. Specifically we will consider the partition of space into disjoint areas and the intersection of such partitions. All are purely geometric considersations and are independent from the associated values.

4.1 Areas

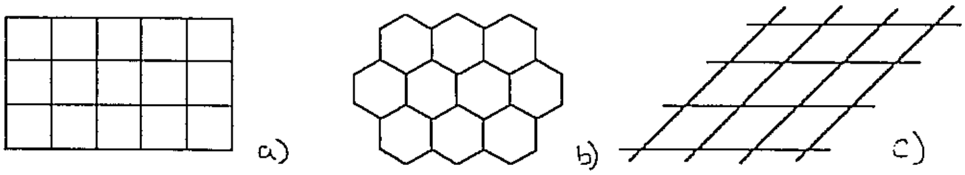
By area we will mean a connected, bounded subset of space (this is different from "regions" in [Tomlin 1983] or the "zones" in [Goodchild 1977]. Area descriptions can be formulated by a bounding polygon or as the aggregation of previously defined areas. Generally the bounding polygon is encoded as a sequence of points which are to be connected by straight lines, but other representations are possible.

4.2 Partitions

Partitions of space are spatial subdivisions (called blocks) constructed so that no two blocks intersect and the sum of all blocks is the total area. Mathematically, partitions are typically created by an equivalence relation, i.e. a relation which is reflexive, symmetric, and transitive (for example: "equal"). All points which have the same property are collected into the same area or block. This reflects the natural assumption that an area designation exists in the first place because its contents somehow have something in common. Partitions can be defined such that all areas with the same value are considered one "region" [Tomlin 1983] or "zone" [Goodchild 1977]; we will only consider connected blocks as areas.

If a spatial subdivision does not form a partition because the areas do not fill the whole space, we can complete it by adding the open space as a defined area (with the property value "null"). If the areas in two spatial subdivisions intersect, a partition is constructed by intersecting all areas and constructing new, smaller areas (which have the same value for all attributes). That is, however, already essentially the solution of an overlay operation.

A special case of a partitioning occurs if the polygons are formed by a regular tesselation of space, i.e. a regular raster, and all areas of interest are described as aggregations of such basic raster cells. It is not necessary that a raster be formed from square fields; other regular tesselations can be used as well (see fig.1) [Diaz 1983] [Samet 1984].



Hierarchies of regular tesselations are commonly used (only a and c in fig. 1 form hierarchies). Areas of interest are described as the smallest collection of units from any level. Best known are quad tree structures [Samet 1984] using a hierarchical tesselation based on squares with doubling side length, as shown in figure ... Such hierarchical structures are more compact descriptions for areas than partitions of a single uniform size since they can adapt to differences in required detail [Lauzon & Mark 1984].

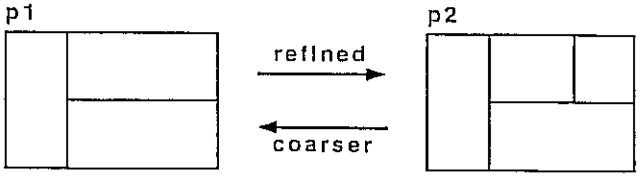
Partitions can be formed by describing each block with a geometric description. Each block can be represented as a closed polygon, but in order to maintain consistency, e.g. with the properties of a partition, a topological data structure is often used [Corbett 1975].

4.3 Partial Ordering Of Partitions Induced By Refinement

A partition, p.2, is said to be a "refinement" of another, p.1, if each block in p.2 is contained in an area of p.1. We can think of a refined partition as the original partition with at least one area subdivided. With this definition of refinement, partitions are partially ordered. Given two partitions, p.a and p.b, it is possible to decide if p.a is a refinement of p.b or p.b a refinement of p.a or neither is a refinement of the other. If both are refinements of the other, they must be equal.

Partial ordering

> a set A (a,b,c...) is partially ordered by a relation greater_than_or_equal_to (>=). This relation is reflexive, antisymmetric, and transitive. Given a and b, either a >= b, or b >= a, or a and b are incomparable such that we can not conclude from 'not(a <= b)' that 'b <= a'



whereas the lattice induced by spatial inclusion is formed by the spatial subdivision in a single block.

4.7 The Most Refined Partition

The lattice of partitions contains a most refined partition (called the infinum) which is more refined than any other partition. The areas or blocks in this partition form the smallest spatial units for which all attribute values are uniform. Such areas have been called Least Common Geographic Units [Chrisman 1975], or Geographic Tabulation Unit Base (GTUB) [Meixler 1985], but the same concept is included in all raster oriented systems where the most refined partition is a regular tesselation, e.g. a square raster. (There is also a least refined partition, which is the undivided universe, called the supremum).

From the lattice structure of partitions with respect to refinement, we know that,

1. if p.0 is a refinement of p.1, and
2. p.0 is a refinement of p.2 (for example, p.0 is the Least Common Geographic Unit partition), then it follows that
3. p.0 is also a refinement of the intersection of p.1 and p.2 [Gill, 1976].

Thus if the partition p.0 is once computed, all geometric intersections can be computed without any geometric operations. Every "coarser" (i.e. less refined) partition is built as set of sets of blocks of the most refined partition. The "geometric" construction of the refined partition becomes purely a set union and intersection operation and no metric operations (like intersection of lines) need be performed. In order to determine the boundaries of the newly formed blocks, a "boundary" operation is applied. [Frank & Kuhn 1986].

The situation where all partitions are constructed from a most refined one is trivially fulfilled with raster representations of areas. Intersection operations in such systems are not difficult. If, however, two partitions must be intersected which are not both formed from the same, more refined partition, geometric constructions to form new, smaller geometric units are required. Such constructions are notoriously difficult.

4.8 Difficulties Of Practical Intersection Computations

Designing programs to compute the geometric intersection of two partitions is difficult and many of the available programs do not properly treat some input configurations. A recent test of several commercially available geometric overlay programs revealed that none worked flawlessly.

First, computers can represent point positions with finite precision only [Chrisman 1984] and this is insuffficent for a complete model of geometry. Because of rounding a point may appear to move from the left side of a line to its right by just rotating or scaling [Franklin 1984]. Inserting new points in lines may slightly change the position of a part of the line and perhaps change established topological relations between points and lines.

Secondly, it may be necessary to detect whether a point in one input set is the same as a point in another set (and similarly for the other topological relations). This decision cannot be made by comparing point coordinates only. Because of the inevitable random errors in measurement or a possibly different lineage for computations in the two data sets, coordinates intended to reference the same points may be quite different. Often points within a short distance (tolerance) are identified appropriately, but such methods use some arbitrary threshold which influences the results. If the tolerance selected is too fine, a large number of small areas (gaps and slivers) appear in the result because it is not detected that points in both input sets mean the same point (or a point in one set is incident with a line in the other set). If, however, the tolerance selected is too gross, areas of importance disappear (e.g. roads and rivers). If two points with different coordinates are identified, new, adjusted coordinate values must be selected. Thus points "move" and can come close to other points, with which they are then further identified. This can lead to substantial changes in point coordinates which are not necessarily correct.

These problems are fundamental and due to the statistical nature of coordinate values (coordinate values are non-estimable quantities). A good program should produce, for any consistent input, a consistent output, perhaps with minimal differences dependent on the order of processing. Correct treatment, however, requires additional information to guide the process.

5 TREATMENT OF NON-GEOMETRIC PROPERTIES

After solving the geometric intersection problem we have to combine the associated data. It is important to note that this step is indepentdent from the geometric operations. We have to split the overlay operation into geometric intersection and non-geometric value combination procedures.

5.1 Thematic Layer

Similarly to the composition of a spatial object from geomtric description and non-geometric properties, we define a (thematic) layer as a geometric partition together with the values for a property. The name of the layer is often the property name and a value (possibly "null") must be associated with every area in the partition.

Building layers from spatial objects can change the focus away from the object view.

(thematic) layer = property name + partition + property value for each area (block) in the partition

In lieu of considering single objects, we see all areas with the same value for the property. A region [Tomlin 1983] contains all areas with the same value, but it is not necessarily connected. In order to keep the object view, a layer must contain, as values, the object identifiers, but then only very limited operations are possible on these values. It seems as if the concept of layers and overlay operations abstracts from the "object" characteristic and concentrate on property values.

5.2 Overlay Operation

One of the most important, if not the most important, operation for spatial analysis is overlaying two (or several) thematic layers. The overlay operation can be dissected into several simpler operations:

5.2.1 Intersection Of Partitions - A common refined partition from given input partitions must be found. If a most refined partition has been computed previously, no additional geometric operations are needed.

5.2.2 Value Distribution - The values from the input layers are distributed to each block in the new refined partition such that all new blocks which together form a block from the input layer have the same value. Each new block gets a value from each input set. This step is trivial if the two input partitions are the same, e.g. the same regular raster.

5.2.3 Value Combination - The two (or several) values for each block in the new common partition are combined to form the desired output value. Some typical operations for combining values are: (weighted) average of the two values, thresholds and Boolean combinations.

It is necessary that the operations be legal for the type of encoding used for the measurement (e.g. it is meaningless to add or substract nominally encoded values, even if they are represented with real values).

The customary reliance on standard data types from programming languages (integer or real numbers, character strings, etc.) avoids the issue; such standard types include all necessary operations and more and lead to abuse. Another often used escape, e.g. in [Tomlin 1983] is to translate all values into real numbers so that most customary operations are available. This, however, may encourage users to improperly attempt to perform operation that make no sense, for instance to calculate averages from two different land-use classes (What is the average of INDUSTRIAL, represented as 3.0, and RESIDENTIAL, represented as 1.0? Certainly not AGRICULTURUAL which happens to be represented as 2.0).

5.2.4 Classification Of Results - The values for the areas or block may contain more detail than desired (e.g. the values are represented as real numbers, but the information desired is on an ordinal scale with three values "low", "medium" and "high"). It has been observed that classification on small nominal or ordinal scales is more useful for decision making than apparently more accurate values on ratio or interval scales. It may that the detail of the values implies a precision which is not truely available. It is then useful to reduce the values to a smaller number by some classification method.

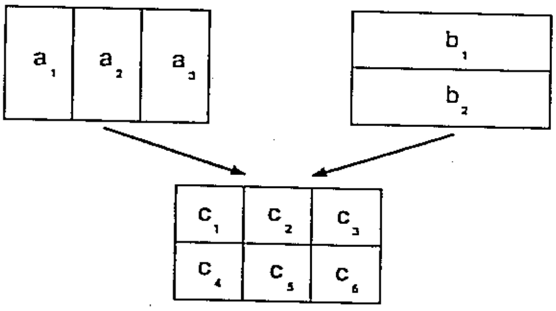
5.2.5 Aggregation - Several connected areas in the resulting partition may have the same resulting value. These areas should be aggregated into one single area. This is not always done, however, often simply because of the difficulty of visually discerning which collection of areas have not yet been aggregated. Nevertheless, aggregation of this kind is necessary for further processing (e.g. to

4.4 Stepwise Refinement Of A Partition

A partitioned space can be refined (with a minimal refinement step) by dividing one block into two new blocks. Similarly a partition can be made less refined (again in a minimal step) by aggregating two adjoining blocks into a single one. With these two simple operations any arbitrary partitioning can be constructed.

4.5 Intersection Of Two Partitions

The intersection operation of two partitions of the same space determines another partition which is a refinement of both. This new partition is the result of the intersection process and can be computed (theoretically) by pairwise intersection of the original areas. The intersecting areas in each original partition can be represented as sets of areas in the new partition.



4.6 Partitions Form A Lattice Structure

As refinement is defined above, partitions form not only a partially ordered set, but a lattice. A lattice is an algebraic structure in which two operations "greatest lower bound" and "least upper bound" are defined for any two elements. The results of the operations are unique for elements involved. The least upper bound of two partitions is the least refined partition which is a refinement for each of the given partitions. The greatest lower bound is, analogously, the most refined partition of which both the given partitions are refinements.

From lattice theory we know that, least upper bound is a commutative operation, i.e. intersecting p.1 with p.2 produces the same result as intersecting p.2 with p.1. The associative law is also valid, i.e. for intersecting a number of partitions, it does not matter if we intersect first p.1 with p.2 and then the result with p.3 or start with intersection of p.2 and p.3 and intersect this result with p.1. Finally the least upper bound is also idempotent, i.e. intersecting a partition with itself produces the original partition [Gill 1976]. Unfortunately the actual implementations cannot achieve this theoretical result and results computed may be quite different depending in which order the intersection operations are executed.

The lattice structure formed by partitions with respect to refinement is different from the partial ordering of spatial subdivisions with respect to "inclusion" (e.g. town A contains parcel 218); the latter structure can also be completed to form a lattice [Saalfeld 1985]. The two concepts are related, but distinct: the lattice induced by refinement is one in which the elements are the partitions, permit the determination of the length of the circumference of areas or to recognize the fact that a area of a given value is completely surrounded by areas of another value).

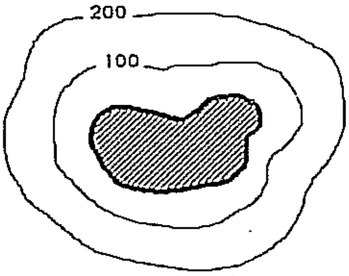
5.3 Application Of Operations On Layers

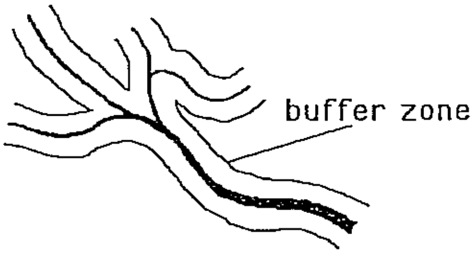
If an operation can be applied to each value in a layer, we can apply the operation to the layer with the meaning that the operation should be applied to each value in the layer (an example is a classification, which maps from real values to some nominal classes). Similarly if an operation can combine two (or more) values from two (or more) layers, we can apply this operation to the layers, again with the semantic that the values referencing the same area should be combined using the given operation (example: computing the average of two or more layers). The results of such operations create a new layer. This is related to the "application" of functions in new functional languages (e.g. Hope [Bailey 1985], or FP [Backus 78] or, with limitations, in APL [Iverson 1962].

Applying operations defined on the values does not include any geometric processing and is usually relatively efficient. Again, it is important to impose the control that only operations which are legal for the given type of values are executed. Such operations, however, are not sufficient for all problems in spatial analysis.

6 CREATION OF LAYERS WITH GEOMETRIC MEANING

For certain operations, overlays with specific values are created, based on geometric operations, e.g. distance from a geometric object.





This may be relative straightforward in a regular raster based system and [Tomlin 1983] gives many examples on how such overlays can be created for irregular partitions. Often a different approach is selected, as when a new partition is constructed which uses boundaries representing specific values for the interested functions (e.g. lines of even 100 foot distances from a pond).

In order to select a method, an effort to understand the information need of the user should be made and then the most appropriate method chosen. In every case, a generalization takes place and some error is introduced; the goal may be to introduce minimal or uniform error to achieve a most equitable result, etc.

7 OVERLAY PROCESSING IN GEOGRAPHIC INFORMATION SYSTEMS

Geographic information processing has in the past been oriented towards mediated batch processing of files. Users would expresse their needs for spatial information and specialists would be employed to produce the information product using the GIS. Increasingly users have been demanding an immediate response, an interactive direct access to the information resource in order to use the available data more effectively and innovatively. Moving from batch oriented, mediated and delayed processing to an interactive situation requires a change in the organization of data.

In order to be effective, the GIS must become easier to use with less effort to learn. Simpler, well structured conceptual models and user friendly interfaces are necessary to reach this goal. It is assumed that the separation into geometric intersection processing and attribute data overlay is useful in this respect.

7.1 Geographic Information Systems Based On Databases

Organization of data becomes important, as the same data is used more often. The database concept where data from multiple sources and with different meanings are logically grouped together and managed by a single software package is appropriate. In advanced systems, multiple users have access to the same data concurrently. Generally, a database management system should contain means to secure data against abuse and accidental loss and it should help to maintain the data consistency.

7.2 Precomputing Intersections

We move from the occasional demand to combine some data files used normally for other purposes to the situation where data are often combined. It becomes thus advantageous to combine and integrate data once and simplify subsequent processing. It may be necessary to secure the help of GIS specialists for the integration, but the users can later retrieve data on their own. For the overlay operation, it is possible to compute the geometric intersection of partitions only once, when data is integrated into the system. The combination of attribute data, however, can not be done ahead of time, as the user's needs are not known. If a spatial data collection is established for long term usage and interactive access for users is demanded, integrating the geometric data and precomputing the intersection of partitions brings the following benefits:

First, the complex and time consuming geometric intersection computations are performed once only. Data integration is generally a time consuming process as consistency checks are made and cleaning of data performed. Additional processing in this phase does not increase efforts significantly. Indeed it may be argued that geometric integration by precomputing the intersections is a necessary part of geometric consistency checking. Eventually, when users pose queries, no additional geometric processing is needed. All overlays can be performed by simply combining the attribute values. A significant saving in computation and in response time must result.

Second, the precomputation of intersections during integration of data opens the possibility to use additional information from the user to resolve dubious cases. We assume that at the time of data integration, users will know more about the quality of the available data. They would beter know the methods used for data collection and the lineage of the data [Chrisman 1983] and are in the best position to select an appropriate tolerances or to decide the identity of points in an interactive dialogue.

7.3 Precomputed Intersection And Data Normalization

Database design uses normalization rules [Date 1983] to design database schema. These rules lead to a systematic break down of data elements (records) to avoid redundancy in various forms. Redundancy is avoided not primarily to reduce the amount of data to store, but to reduce the possibility of inconsistencies and problems during changes (anomaly of update).

Precomputing the geometric intersection can be considered similarly: redundancy is reduced since all common boundaries of all areas are recognized and stored only once. If the data reference the same boundaries often, a situation typically found in large scale geographic data, not only an improvement in performance, but also a reduction in storage may result. More importantly, for any two points or lines the identity problem is resolved and the data processing can use the "unique name" assumption [Reiter 1984].

The overlay of two different layers, both expressed as values with references to the areas of the most refined partition, is primarily a "join" (exactly an "equi-join") using the common reference to the area and efficient methods for execution are known in database literature [Ullman 1980].

8 FUTURE WORK

A number of topics for work have been touched on:

Consistent and formal definition of legal operations for interval, ordinal and nominal data (each extended with values for "unknown", "not applicable", etc.) should be worked on. This would allow for the integration of knowledge about meaningful combinations of values into an GIS and result in advice to the user if inappropriate operations are tried.

Spatial analysis demands more operations than the standard set of arithmetic operations. In [Tomlin 1983] a large number of methods for the creation of rasters filled with values with geometric meaning are given. It would benefit systems which are not raster based to include similar operations (e.g. the "buffer" in ARC/INFO). A systematic study could help to better understand spatial analysis operations and would certainly improve user interfaces and implementations.

Processing of layers does not stress the "object" characteristic, but prefers the "region" approach. Spatial analysis, however, requires both approaches. It is difficult to formlulate queries like: "Find all school districts which contain more than 4 school buildings." In an overlay oriented system. It would be interesting to understand the limitations of each method and see how they can be integrated.

We propose here that the most refined partition is precomputed during the integration of geometric data. This may result in a large set of relatively small areas and spatial objects are then composed of a large number of such small areas. To improve performance, operations on larger objects will be necessary. We assume that application of theories on partially ordered sets and lattices will lead to solutions here.

9 CONCLUSIONS

We have separated a geometric and a non-geometric part in the overlay processing in spatial information systems. Overlay processing is based on layers of data where a given property a value is associated with an delimited area. It is useful to complete the areas to a partition, such that they fill the space completely and do not overlap. Overlaying two layers consists of computing the geometric intersection of the defined areas and of combining the values for the areas.

The geometric operation of intersection of two partitions is difficult to implement on computers and a number of special situations must be dealt with correctly. The problems are due to the limited precision with which point coordinates can be represented. Furthermore, coordinate values for the same point, but from different sources, do not usually agree and identification of similar points in two data sets are difficult. We propose that geometric data be integrated into a system by computing the most refined partition, i.e. compute the intersection of all available data sets. This has the advantage that subsequent processing is simplified. If a GIS is built for a long term usage with interactive access to the data by the users of that data, performing some difficult and time consuming operations like geometric intersection of partitions only once in preparation for quick responses is effective.

Geometric integration of data by computing the intersection can be seen as a form of "normalization" of the data, as common points and lines are recognized and multiple storage reduced. Combining spatial data which is referenced to this most refined partition becomes a non-geometric database join operation.

Operation on single data values can be applied to geographic layers, which contain values for a specific property together with references to the areas in the most refined partition. Such an application of an operation has the meaning that values from the same area are operated on individually and the result again associated with this area. This concept of applying an operation which is defined on a single value to a set of values is similarly found in modern, functional programming languages. It separates the operations on the values from the mechanism necessary for the distribution of the single operation over all the values.

The insight gained by the theoretical analysis of overlay processing is directly applicable for the implementation of new GIS. It improves the design of the system and the coding of its operations. More importantly, the separation into a few relatively simple concepts can benefit the design of the user interface and would result in systems which are easier to learn and easier to use.

10 REFERENCES

Backus, J., Can programming be liberated from the von Neumann style? a functional style and its algebra of programs, Comm. ACM, Vol. 21, No. 8, Aug. 1978

Bailey, R., A Hope tutorial, BYTE Magazine, vol. 10, no. 8, Aug. 1985

Carlson, E., Three dimensional conceptual modeling of subsurface structures, Proc. ASPRS-ACSM Conference, 1987

Chrisman, N.R., On storage of coordinates in geographic information systems, Geo-Processing, Vol. 2, 1984

Chrisman, N.R., Concepts of space as a guide to cartographic data structures, in Dutton, G. (Ed.), First International Advanced Study Symposium on Topological Data Structures for Geographic Information Systems, Harvard Papers on Geographic Information Systems, Harvard University, Cambridge, Mass., 1978

Chrisman, N.R., Topological information systems for geographic representation, Proc. AUTO-CARTO 2, Reston, VA, 1975

Chrisman, N.R., The role of quality information in the long-term functioning of a geographic information system, Proc. AUTO-CARTO 6, 1983

Codd, E.F., Missing information (applicable and inapplicable) in relational databases, SIGMOD Record, vol. 15, no. 4., December 1986

Corbett, J.P., Topological principles in cartography, Proc. AUTO-CARTO 2, Reston, VA, 1975

Date, C.J., Null values in databases management, Proc. 2nd British National Conference on Databases, Bristol, England, 1982

Date, C.J., An Introduction to Database Systems, Vol 1, 3rd edition, Addison-Wesley, 1983

Diaz, B.M., Bell, S.B.M., Holroydt, F., Jackson, M.J., Spatially referenced methods of processing raster and vector data, Image and Visual Computing, vol 1, no 4, Nov., 1983

Frank, A.U., Kuhn, W. Cell graphs: a provable correct method for the storage of geometry, 2nd International Symposium on Spatial Data Handling, Seattle, 1986

Frank, A.U., A conceptual framework for land information systems: a first approach, Report 38, Surveying Engr. Dept., University of Maine, 1984

Franklin, W.R., Cartographic errors symptomatic of underlying algebra problems, Proc. Internat'l Symposium on Spatial Data Handling, Zurich, 1984

Giblin, P., Graphs, Surfaces, and Homology, Chapman and Hall, London, 1977

Gill, A., Applied Algebra for the Computer Sciences, Prentice Hall, 1976

Goodchild, M.F., Ross, J.H., Swanson, W.G., PLUS: a conversational regional planning tool, Lands Directorate, Fisheries and Environment Canada, Ottawa, 1977

Guttag, J., Horning, J.J., Formal specification as a design tool, ACM Symposium on Principles of Programming Languages, Las Vegas, 1980

Guttag, J., Abstract data types and the development of data structures, Comm. ACM, June 1977

Guttag, J., et al., The design of data specifications, in: Yeh, R.T. (Ed.), Current Trends in Programming Methodology, Vol. 4, Data Structuring, Prentice-Hall, 1978

Iverson K.E., A Programming Language, Wiley Publishing Co, 1962

Mark, D.M., Lauzon, J.P., Linear quadtrees for geographic information systems, Proc. Internat'l Symposium on Spatial Dta Handling, Zurich, 1984

Meixler, D., Storing, retrieving and maintaining information on geographic strucures, a Geographic Tabulation Unit Base (GTUB) approach, Proc. AUTO-CARTO 7, Washington DC, 1985

Parnas, D.L., A technique for software module specification with examples, Comm. ACM, Vol. 15, No. 5, May 1972

Reiter, R., Towards a logical reconstruction of relational database theory, in: Brodie, M.L., et al. (Eds), On Conceptual Modelling, Perspectives from Artificial Intelligence, Databases, and Programming Languages, Springer Verlag, New York 1984

Saalfeld, A.J., Lattice structures in geometry, Proc. AUTO-CARTO 7, Washington DC, 1985

Samet, H., The quadtree and related hierarchical data structures, Computing Surveys, Vol. 16, No. 2, June 1984

Shipman, D.W., The functional data model and the data language DAPLEX, ACM Transactions on Database Systems, Vol. 6, No. 1, March 1981

Stevens, S.S., On the theory of scales of measurement, Science Magazine, vol. 103, 1946

Tomlin, C.D., Digital cartographic modeling techniques in environmental planning, Ph.D Thesis, Yale Univ., 1983

Ullman, J.D., Principles of Database Systems, Computer Science Press, Potomac MD, 1980