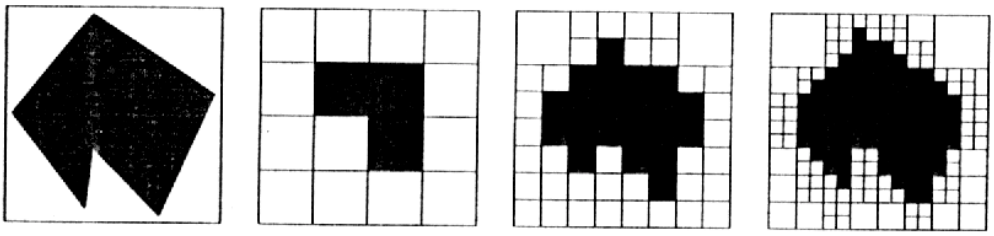
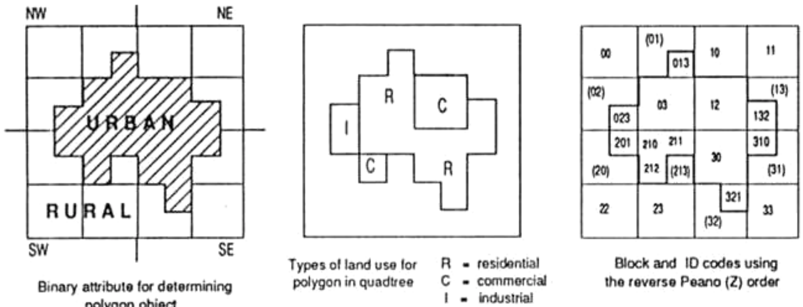
Variable Spatial Resolution: Quadtrees

The concept of variable spatial resolution implies varying sized units at a given resolution level. The choise of the shape is a special matter. The square is particular handy if the process of creating the blocks of varying size is one of the decomposing space from a general level to more detail. For example, a polygon can be successively approximated by sets of blocks at different levels:



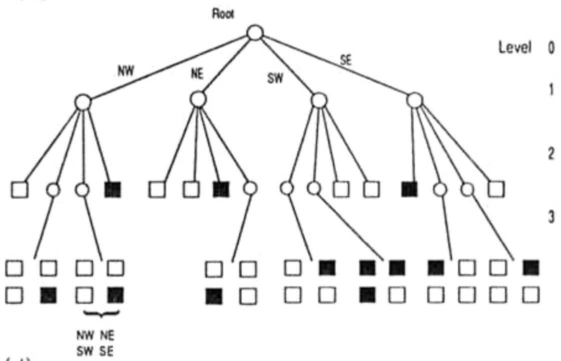
If the process involves systematic splitting of space in 2-dimensional space by a rule of four, then the structure is known as a quadtree, a type of hierarchical data model. A 3-dimensional equivalent is known as an octtree because it involves an eightfold splitting:



(a) Binary attribute for determining polygon object

(b) Types of land use for polygon in quadtree — R = residential, C = commercial, I = industrial

(c) Block and ID codes using the reverse Peano (Z) order



(d)

Quadtree Level 1	2	3	Land use type
		013	R
		023	I
	03		R
	12		C
		132	R
		201	I
		210	R
		211	R
		212	C
	30		R
		310	R
		321	R

(e)

Level 2	Level 3	Type of land use	Area (number of level 3 blocks)	Level 2	Level 3	Type of land use	Area (number of level 3 blocks)
00		Rural	4		202	Rural	1
01	010	Rural	1		203	Rural	1
	011	Rural	1	21	210	Residential	1
	012	Rural	1		211	Residential	1
	013	Residential	1		212	Commercial	1
02	020	Rural	1		213	Rural	1
	021	Rural	1	22		Rural	4
	022	Rural	1	23		Rural	4
	023	Industrial	1	30		Residential	4
03		Residential	4	31	310	Residential	1
10		Rural	4		311	Rural	1
11		Rural	4		312	Rural	1
12		Commercial	4		313	Rural	1
13	130	Rural	1	32	320	Rural	1
	131	Rural	1		321	Residential	1
	132	Residential	1		322	Rural	1
	133	Rural	1		323	Rural	1
20	200	Rural	1	33		Rural	4
	201	Industrial	1				

(f)

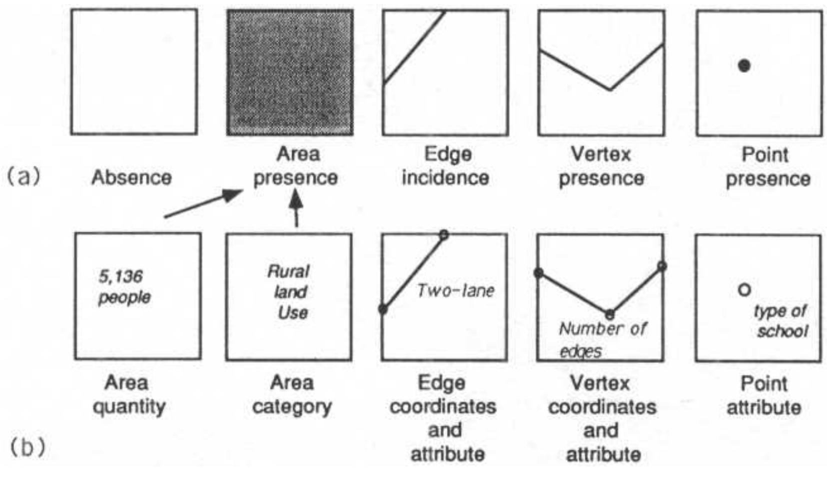
The quadtree data organization, as presented above, leads to complete table of data for the entire area, involving quadtree blocks and identifiers based on location codes, hierarchical organization, table of attributes for the urbal polygon, complete data.

Different kinds of data can be treated in this hierarchical subdivision fashion. Ideally we would like to be able to:

1. treat point, line, and area data in the same way;
2. capture metrical details for entities;
3. facilitate various kinds of operations;
4. deal with different ways of measuring attributes;
5. have consistent locational referencing.

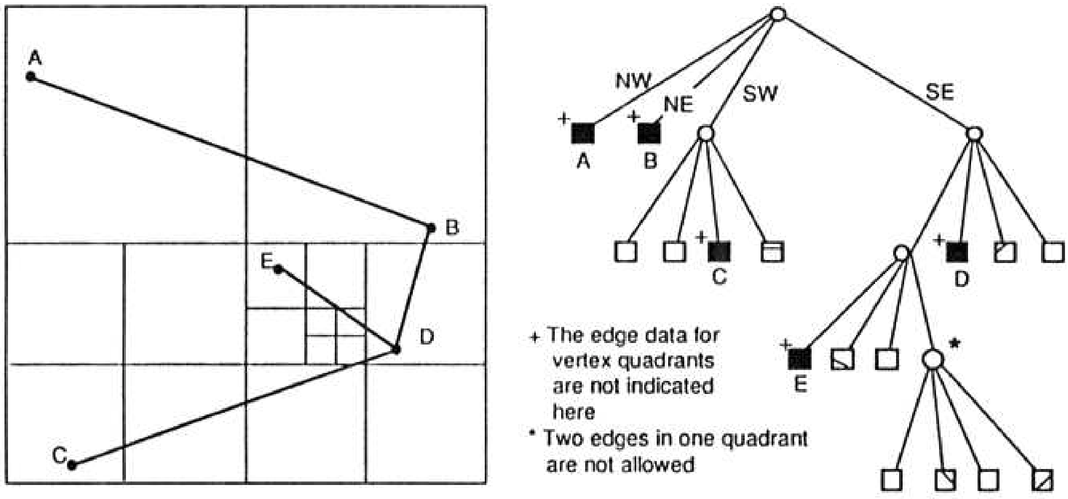
The simplest form of quadtree recognizes the presence or absence of an attribute in space, whether point, line, or area. Computer scientists usually refer to the **binary incidence** representation as colour coding, using black and white to indicate presence and absence.

A cell could contain a scalar value, or a pointer to sets of attributes under the condition that the cell is a lowest geographic unit. Thus cells may be used to represent point data, such as cities, where each cell contains one city; or linear features, say water pipes, where each cell contains a segment of a pipe or a junction of several water lines. So we may define an attribute presence quadrant, an absence quadrant, an edge quadrant, a vertex quadrant and a point quadrant:



In the standard form, the geometry of edges and points is not retained, only incidence. However, as for fixed resolution regular tessellations, additional information can be encoded for cells (b). In the case of edges representing polygon boundaries or graphs, this could consist of the *x* and *y* Cartesian or the polar coordinates to establish where the edges cross the boundary of a cell, or for vertices or points, the exact coordinates for a point within a cell.

If, for linear features, the incidence is only a vertex with one or more graph edges or a piece of an edge, then a quadtree representation will look like that in:



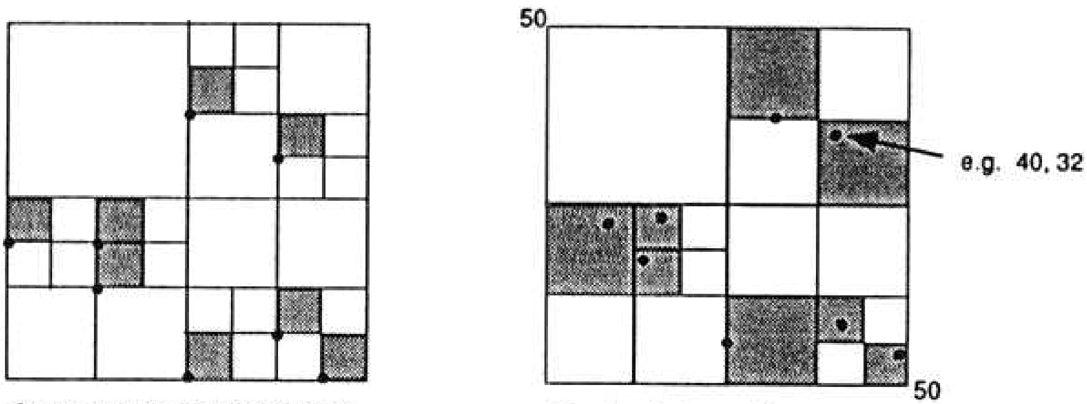
(a) (b)

(c)

Quadrant ID	Vertex label	Vertex coordinates	Edge coordinates	Edge attributes

This figure represents a quadtree for line data, giving map and quadrants for line objects, tree and encoding rules, data item records. Various possibilities for encoding exist; rules must be established before the quadtree database is created from the original data.

Unconnected points may be handled in different ways. A regular figure decomposition process could produce squares from the orthogonal coordinate space by subdividing using both *x* and *y*, with varieties depending on whether or not all four squares at a given level of decomposition were recognized (the MX quadtree) or not (the PR quadtree), The second type requires coordinate information to establish position within the block; the former does **not,** representing the point at a corner of the cell.



(a) Quadrants formed by presence of points; points represented by SW corner of a block

(b) Quadrants formed by presence of points; coordinate data retained (PR quadtree).

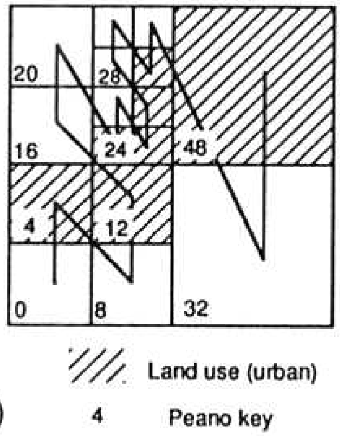
Various locational reference schemes are possible, and are indeed used for meeting different requirements.

Node level	Nodes at next level NW	NE	SW	SE	Node type	Node attributes

(a)

Locational code	Size of quadrant	Node type	Node attributes

(b)



(c)

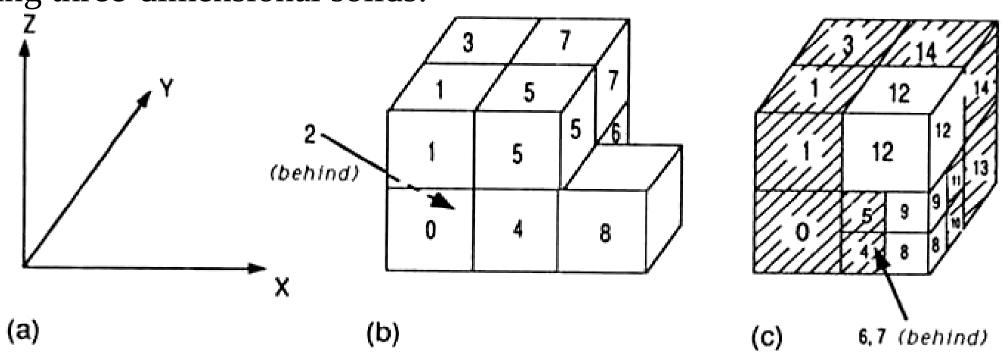
Peano key	Size of quadrant - side length	Land use
0	2	Rural
4	2	Urban
8	2	Rural
.	.	.
.	.	.
.	.	.
28	1	Rural
29	1	Rural
30	1	Urban
31	1	Rural
32	4	Rural
48	4	Urban

(d)

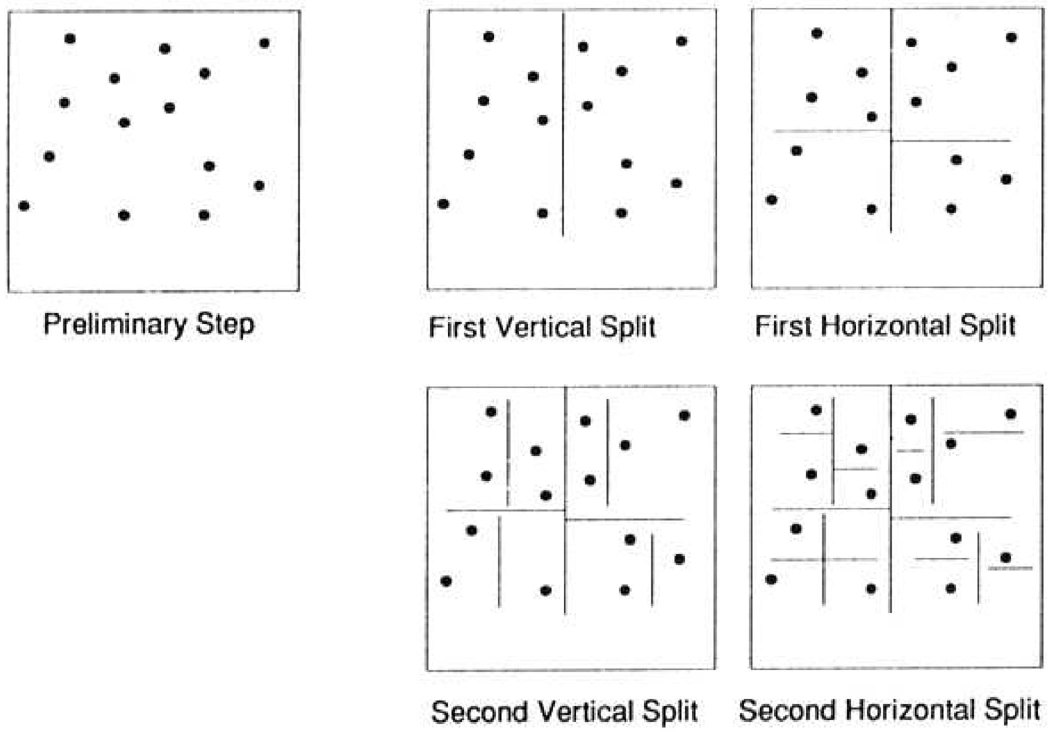
One simple scheme is to consistently order the four blocks at each level in a NW, NE, SW, SE sequence, using data in the record for a tree node to point to the four nodes at a lower level if such exist (a). Numerical coding representing the NW, NE, SW and SE by integers could be used (b); some coordinate values could be used, or a space path could be employed to simplify movement through the entire set of cells, without using actual coordinate values.

Referring to figure above, coding using row and column identifiers would require more data to be stored than for a locational coding scheme, using the NW, NE, SW, SE orientations, while a Peano N path has single dimension addressing and has stable numbering across different levels of resolution. Thus the larger blocks in the quadtree would be represented by fewer positional pieces of data than the number of blocks (c), and the final table would contain items for the Peano key and quadrant size, often the number of smallest size pixels on the side of the square block.

The general properties and principles for quadtrees are applicable to the three-dimensional variant, the octtree, used to some extent for geologic modelling and representing three-dimensional solids:



Hierarchical decompositions may be undertaken on the basis of the **empirical information** to be encoded and stored in contrast to the regular subdivisions so far discussed. The latter are data independent; the former are data dependent. For example, a distribution of point features, such as cities, may be subdivided into rectangular, rather than square blocks, on the basis of alternating x *and* y axes. A similar process can produce two or four branches at each step. Thus, the empirical information, the exact position of the points, governs the data structuring, not a fixed-grid scheme. The binary subdivision, which is one of a group of **K-dimensional (KD) trees,** is generally regarded as superior to the point data quadtree for operations done in sequence.



The quadtree and related structures, clearly based on a tessellated discretization of space, provide semantic value by their recognition of varying density of incidence of phenomena in space, and can deal with both vector and raster data. The hierarchical structuring cleverly addresses spatial variations at different scales, it offers the valuable adaptability property to empirical conditions and with good locational referencing provides a basis for efficient spatial access and indexing. As discussed later, Boolean operations such as union, intersection and difference are easy to perform, whereas translation, rotation and scaling are not.

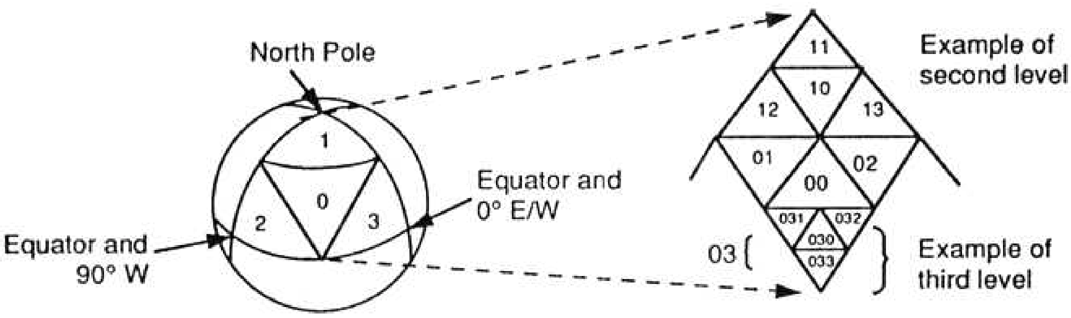
In general, the hierarchical tessellations are regarded as offering benefits in the reduction in the amount of space needed to store data for phenomena. We contrast the more extensive grid cell encoding with the quadtree, and another device, the run-length encoding. The first of these records data for each cell, demonstrated here for two different resolutions. The quadtree will use a smaller number of spatial units as produced by the hierarchical subdivision; the run-length encoding reduces data storage by recording runs of like conditions for rows (or columns) as shown. The degree to which **the space-saving methods reduce storage** depends **primarily on the amount of homogeneity** in the mapped data. The extremes are a perfectly uniform landscape, for which the quadtree block is best, or a checkered pattern in which each cell is different from all its neighbours. In this case, there is no particular advantage in using the two space-saving techniques. Alternative data storage schemes like linked lists are preferable for sparse matrices.

The hierarchical structures may be differentiated on the basis of types of data, the principles guiding or governing the decomposition process, and the type of spatial resolution. However, because they are based on regular spatial units, they also have advantages and limitations associated with the use of grid squares. Particularly, there are limitations in dealing precisely with point and linear features, and in not explicitly addressing topological spatial properties.

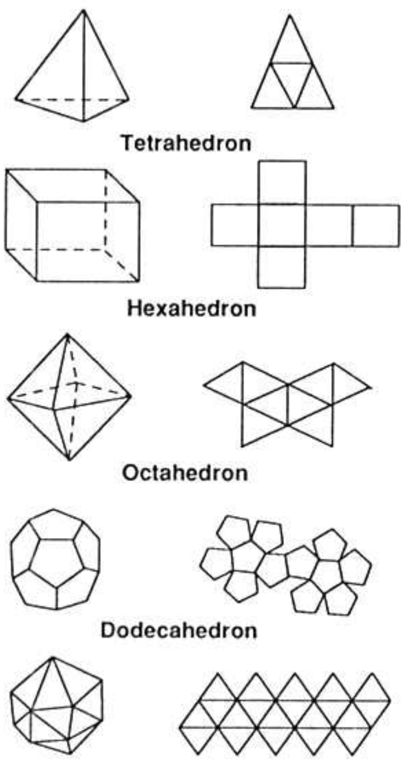
Hierarchical Tessellations for a Sphere

A particularly interesting and challenging need is a hierarchical structure of cells for covering the entire world. As discussed, a global database can be structured as a set of triangles. While there are many ways to subdivide the surface of a ball, at one level, some of the geometrical properties of particular figures may have practical shortcomings for a single or multiple scale representation scheme for the earth.

A globe can be divided into a single tessellation of only triangles that at one level have five corner neighbours, that is the icosahedron threedimensional model. However, it is not possible to anchor the graticule to all important global features, the pole, equator and meridians. The dodecahedron, a set of pentagons, while having a nice property of only touching at edges of cells, is even worse for not having hemispherical symmetry, being flat at the top (or bottom) and pointed at the diametrically opposite pole. Out of the five convex regular polyhedra are shown below, and known as the **Platonic solids.** The icosahedron is the most nearly spherical because it has the largest number of vertices, edges and facets.



The tetrahedron, or pair of these that make the octahedron, is better for fitting to the requirements of polar symmetry and for mapping vertices along the equatorial plane. Six anchor points, each with four triangles meeting, correspond to the north and south poles, and the 0^0, 90^0, 180^0, and 270^0 subdivisions around the globe. The eight initial triangular facets are then subdivided into a set of regular triangles (a), providing global referencing for areas closely similar in shape and size, and facilitating hierarchical referencing to detailed positions on the earth. Within each principal triangle, four equal subdivisions are referred to by 0, 1, 2 and 3; at greater resolution levels there will be extra digits, using this same scheme of four numbers (b). The numbering scheme used for the tetrahedron can also be regarded as providing indexing keys for access to each cell.



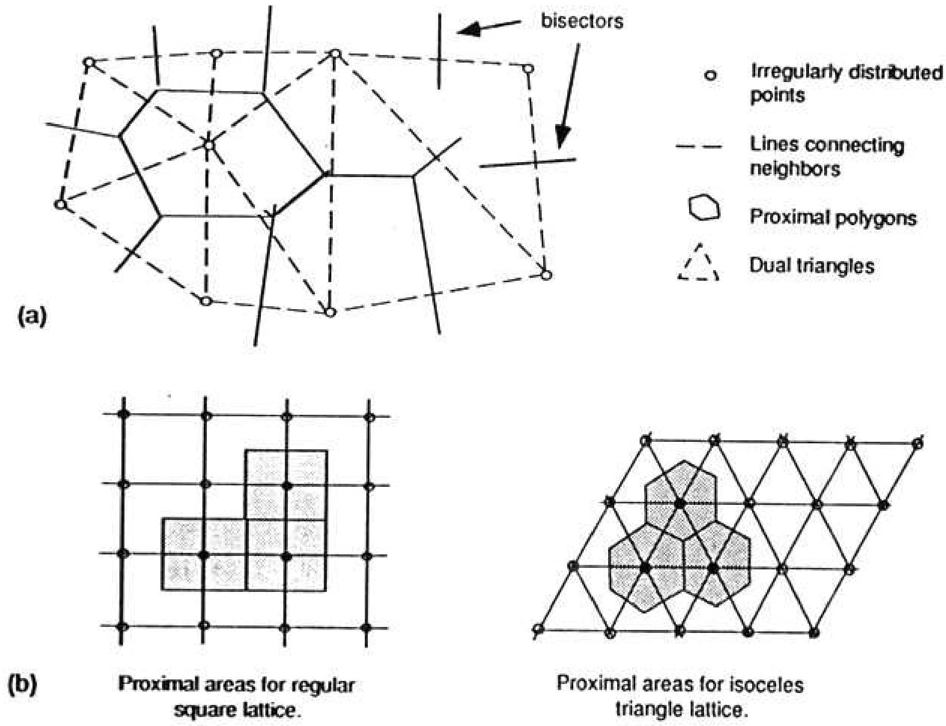
Three dimensional view / Developed planar view	Edges per face	Faces per vertex	Vertices	Edges	Faces
Tetrahedron	3	3	4	6	4
Hexahedron	4	3	8	12	6
Octahedron	3	4	6	8	12
Dodecahedron	5	3	20	30	12
Icosahedron	3	5	12	30	20

Irregular Tessellations Based on Triangles

Two types of irregular tessellation have valuable properties for spatial information systems: triangles and proximal polygons. They both represent variable spatial resolution at a given scale and can be dealt with hierarchically although, at the moment, there are few practical applications of this variety. A real need that demonstrates the value of a set of triangles is the representation of earth surface terrain conditions. It is generally thought that, at least visually, it is preferable to break up a surface into triangular facets rather than squares or other polygons. In order to create areal units from only point data, a technique of creating proximal polygons is often used.

Proximal regions

For the second of these needs, consider a distribution of administrative offices in space. We can argue for locating them so that their territories are demarcated such that the people in every household living within them travel to their nearest centre:

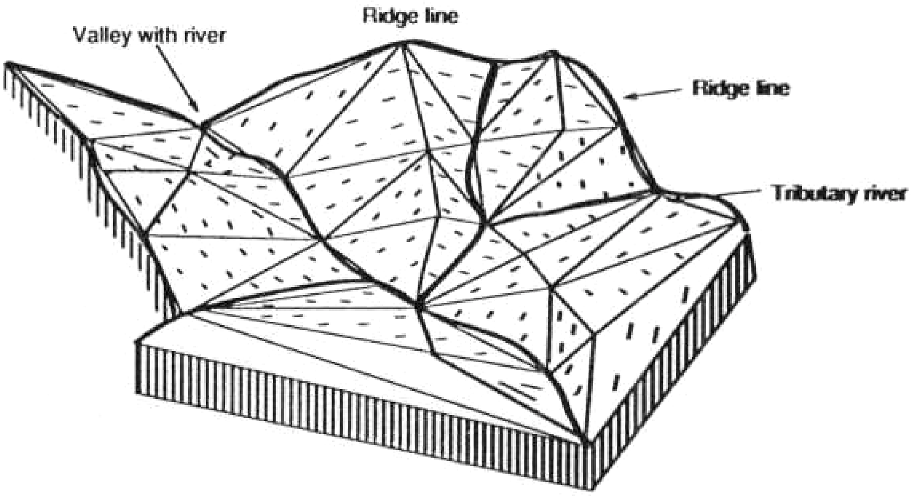


As such, the polygons are sometimes called proximal regions, or are often called **Thiessen** or **Voronoi polygons.** If any people in a household within the designated proximal zone then choose to travel to a different centre, they bear an additional cost by increasing their travel distance above an optimum minimum based on the nearest centre.

The polygons are created by subdividing lines joining nearest neighbour points, drawing perpendicular bisectors (sometimes called mediators) through those points, and then assembling the several polygon edge pieces out of those lines, as shown (a). The concept of the proximal area, sometimes used as a standard for evaluating equity issues for travel to administrative centres or public service facilities, is known in mathematics as the **Dirichlet domain.** This space encompasses a set of points closer to a given point than to any other points in the set. With reference to (b), note that the domain takes on different forms for varying point patterns, regular or otherwise.

Triangulation

The irregular triangulation for surface modelling is a somewhat more involved concept than creating proximal areas because it is oriented to line features as well as points. As figure below suggests, using triangles to represent terrain, a more realistic representation will be achieved if the spatial data units recognize natural surface changes in slope, at peaks, pits, passes, ridge lines, saddle points and course lines or discontinuities, rather than just be fitted arbitrarily. A set of triangular facets can be created to meet these conditions by having triangle edges fall along approximations of ridges and river channels, and having their corners located at control points with exact known coordinates from earth surveys, or at river confluences, or at peaks or depressions of terrain.



Modelling terrain: concept of a triangulation closely following the major terrain features.

Ideally we would like to have:

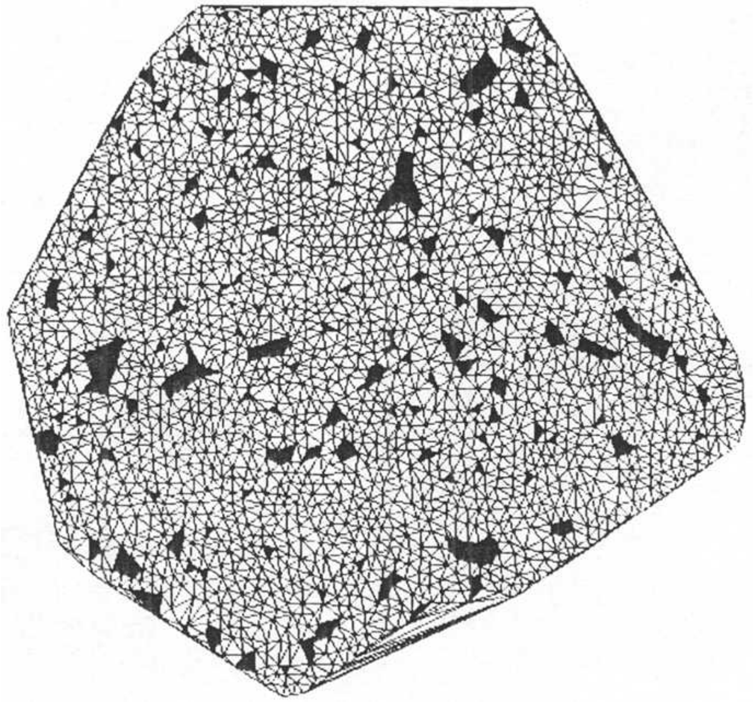
1. the triangle corners match important turning points in the terrain surface;
2. the important linear features be represented by triangle edges.

The process of triangulation has three stages:

1. choosing the data points;
2. connecting points to create triangles;
3. storage of necessary and additional desirable information.

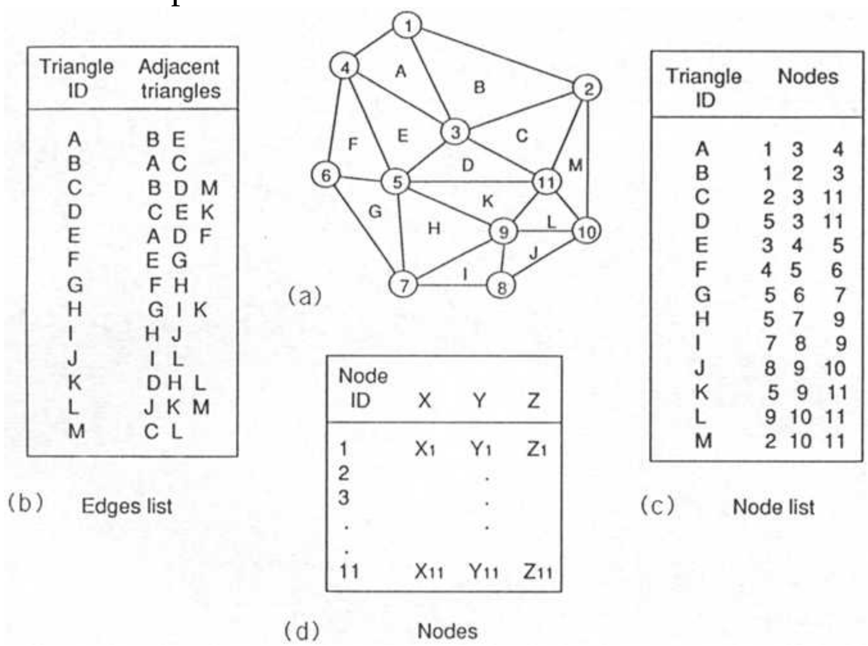
Assuming for the moment that the z variable is terrain elevation, but noting that, in principle, other phenomena can be treated in the same manner, then original data for height may come from several sources of different spatial structure. Data may be lattices of heights in the form of digital elevation models (themselves often created from digitized contours), irregularly distributed spot heights, contours or a mixture. It is important to know the spatial distribution and whether data are point or linear. It is also most important, especially for terrains, to recognize natural breaks of slope, and key landscape features like coastlines, course lines, ridges and peaks.

Whatever the original form of data, the triangulation method uses *x, y,* z coordinate triads, fitting a set of irregular triangles to all data points, and then interpolating intermediate values of z from the known values at the corners of the triangles. This **triangulated irregular network** (TIN) therefore is a tessellation model applied to known positions, or, at least a subset of them. Because estimates of height will be more reliable the closer they are to the original data points, it is better to use triangles as close to equilateral as possible. It is especially important to avoid long narrow triangles, such as might occur when using data from widely spaced contours with many points on each contour line.



TIN for high density of data points. The black areas indicate triangles with virtually no slope. The hull of outer points is not necessary convex. Thus the triangular tessellation consists of:

1. a set of points carrying elevation data;
2. a set of lines consisting of pairs of points, joined by straight lines;
3. a set of triangles, having triplets of *x*, *y*, *z* coordinates;
4. adjacency relations for the edges of the triangles;
5. a list of triangles in which particular edges are included;
6. the triangles in which particular nodes are contained.



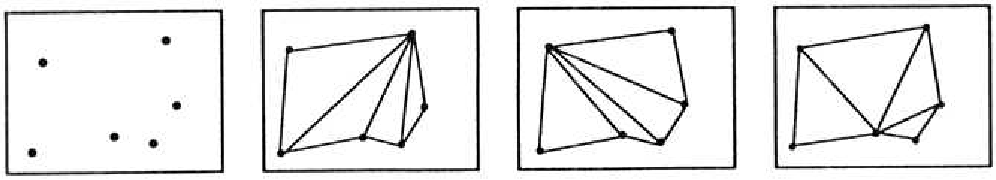
Triangle ID	Adjacent triangles
A	B E
B	A C
C	B D M
D	C E K
E	A D F
F	E G
G	F H
H	G I K
I	H J
J	I L
K	D H L
L	J K M
M	C L

Triangle ID	Nodes		
A	1	3	4
B	1	2	3
C	2	3	11
D	5	3	11
E	3	4	5
F	4	5	6
G	5	6	7
H	5	7	9
I	7	8	9
J	8	9	10
K	5	9	11
L	9	10	11
M	2	10	11

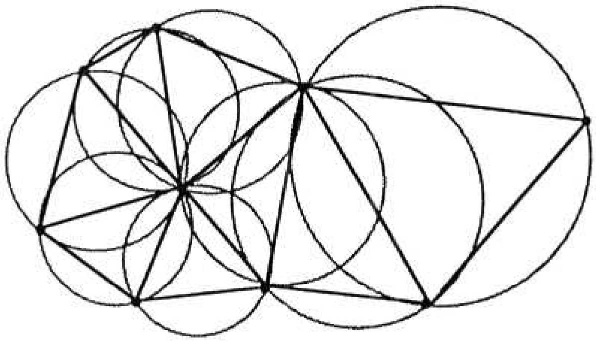
Node ID	X	Y	Z
1	X_1	Y_1	Z_1
2		.	
3		.	
.		.	
.			
11	X_{11}	Y_{11}	Z_{11}

As such the triangular tessellation combines topological and geometric information. Various derived data are computable as needed or computed and stored for the following topological elements:

1. the gradient and aspect of edges of triangles;
2. the planar and surface area of the triangular sets;
3. the slope and aspect across the facet.



(a) A set of points and three instances of a triangulation of these points. The third is closest to the ideal of equilateral triangles.



(b)

Naturally many sets of triangles could be fitted to a set of points (a). A triangulation that produces **Delaunay triangles** is generally the preferred relatively straightforward method, producing triangles with a low variance in edge length. This type of triangle, based on a proximal distance criterion, is defined by the condition that the circumscribing circle of any triangle does not contain any point of the data set inside it (b). The Delaunay triangles are duals to the Thiessen polygons, and the polygon centre is the centre of the circumscribing circle.

Triangulated tessellations have a number of useful features for spatial information systems. The triangles can be treated as irregular polygons; the tessellation exhausts all space; there is planar enforcement; they are appealling spatial units that appear to provide acceptable models of certain kinds of terrain surface. At the same time, their creation is computationally demanding; there are many possible triangulations for any set of points; and they can miss important aspects of surface morphology unless the edges are constrained to fit major breaks of slope.

They are not the only data models conceptually possible or actually used in estimating surface height values from a set of spatially distributed points. If contour data exist, elevations can be interpolated for points lying within the isolines. A uniform distribution of points, a lattice form or intersections of grids, facilitates estimation of **heights** of points along the grid square sides, but generally the regular distribution of point values is itself estimated from an irregularly spatially distributed set of data by a process known usually as **gridding.** In simple form, this consists of using one or more original data points found in a neighbourhood of the grid intersection.

Absent much empirical evaluation of different surface representation techniques, it appears from logical grounds and some experimental studies that irregular tessellations can recognize important surface conditions, provide data for topological properties, produce reliable interpolated values by passing the surface through known data points, and allow for different scale representations. Gridding procedures producing regular tessellations do not usually recognize data points, do not provide explicit topological information, and are not adjusted to known conditions like breaklines.

Nevertheless, choices must be made on the basis of purpose and type of terrain being modelled. Some hydrological simulation models work reasonably well with gridded data; subsurface depth estimations from limited information can be done better by grid data for there is usually no information as to natural break conditions. Simple, regular, nearly plane surfaces are better handled by grids, but dissected fluvial landscapes are apparently successfully modelled by triangulated irregular networks. Glaciated landscapes may indeed be best represented by neither technique, but instead by fractal geometry.