

A TAXONOMY OF 2D SPACE TESSELLATION

Y. C. Lee and Z. L. Li

Department of Land Surveying and Geo-Informatics
 The Hong Kong Polytechnic University
 Hung Hom, Kowloon, Hong Kong
LSYCLEE@POLYU.EDU.HK
LSZLLI@POLYU.EDU.HK

ABSTRACT

When we map an area or create a digital database for it, the first task is often to partition the space into smaller units. There are traditionally two methods of partition called vector and raster. A vector partition delineates the boundary of features by lines while a raster partition subdivides space into a regular matrix of square or rectangular pixels. The two partitions are complementary methods of subdividing space either by feature or by arbitrary space cells. With advances in data modeling, variations of the two traditional methods have been developed, such as the representation of a feature by pixels and not by vector lines. At present, there is a lack of terminology to describe the various methods of tessellation.

In this paper, we will propose a taxonomy for two-dimensional space tessellation. Its essential feature is to distinguish between a conceptual model for tessellation and a set of geometric primitives used to implement the tessellation cells. This allows us to systematically describe the various structures resulting from combining an abstract tessellation with different geometric primitives.

1 INTRODUCTION

In order to manage two-dimensional space more effectively, we have traditionally broken it up into discrete points or partitioned it into non-overlapping cells. We call such a partition a tessellation, and the partitions represent either geographic features or space cells that are often arbitrarily defined.

The first type has been used in producing line maps where the extent of geographic features are delineated by lines. The attributes of these features are then represented by annotations, line patterns, or area patterns.

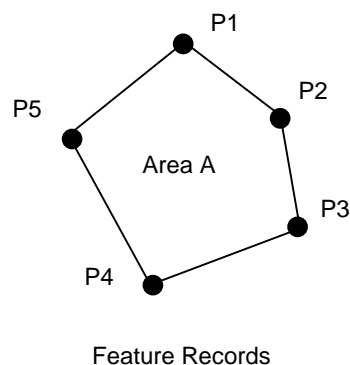
The second type has been used in producing orthophotos or similar image-based products. In this case, the image is broken into arbitrary units such as light-sensitive grains on a photograph or screen dots on a printed version of it. The attributes of these space cells are then represented by size or colour.

We will call the first type a *feature-primary tessellation* (FPT) and the second a *space-primary tessellation* (SPT) of space. They result in *feature cells* and *space cells* respectively.

In digital mapping, the feature or space cells are the basic units of storage. In a database created with FPT, the data for each feature will be logically stored together in one or several linked records. In the so called vector maps, data that are stored logically together for a feature usually include its outline as a series of ordered points and other attributes (Figure 1). In a database created with SPT, on the other hand, the data for each space cell will be logically stored together. In the so called raster maps, the pixels are stored together with their attributes.

It is important, therefore, to note that in either tessellation, both the space and its attributes are stored. The main difference between FPT and SPT is that in the first case, emphasis is placed on the space occupied by individual features. There is a one-to-one relationship between a feature cell and a feature. In the second case, emphasis

is placed on space cells not directly related to the features. The fact that a space cell falls upon a feature is more coincidental than intentional. If a space cell is large, it could fall upon many features. Even when the space cells are very small, the same tessellation can be placed over different themes of data, causing each cell to be associated with multiple features, one from each theme. Hence in a SPT, there is a one-to-many relationship between space cell and features.



Feature ID	Outline	Other Attributes
Area A	P1, P2, P3, P4, P5	

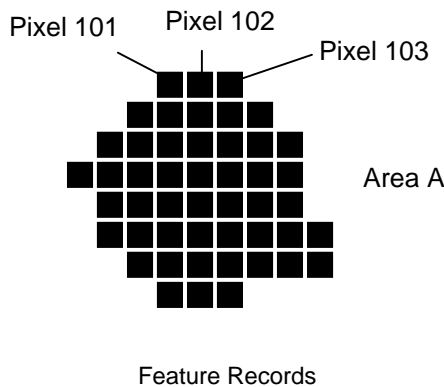
Figure 1: A Vector Representation of Area A

In digital mapping, it has been customary to call maps created by a FPT vector maps and those by a SPT raster maps. We will argue in this paper that this is incorrect, and we will introduce a taxonomy of tessellation that will clarify a number of important concepts. In this discussion, we are not concerned with the representation of space by discrete points such as the case of digital elevation models. Very often, these discrete point representations will be processed to eventually form tessellation.

2 VECTOR AND RASTER

The comparative merits of vector and raster models and the conversion between them have been the center of research for a long time [Peuquet, 1981]. In a vector model, we assume a vector between every two consecutive points to form continuous lines. It is mathematically a piecewise linear interpolation for lines. On the other hand, a raster model uses a matrix of regular pixels that are usually square.

It would appear that the vector model is equivalent to the FPT and the raster model is equivalent to the SPT. However, it is possible and even desirable to represent a feature using a series of pixels instead of a series of points making up its outline. The feature itself will be stored as a unit, and the inside of the feature will be filled with pixels, thus providing the best of the two worlds (Figure 2).



Feature ID	Pixels	Other Attributes
Area A	101, 102, 103, ...	

Figure 2: A Raster Implementation of Area A

On the other hand, it is theoretically possible to regard each pixel as a polygon and represent it in vector format (Figure 3). We do not do this mainly because it is highly inefficient. Indeed, there are irregular space-primary tessellations, such as Triangulated Irregular Networks (TIN), which are stored in vector format.

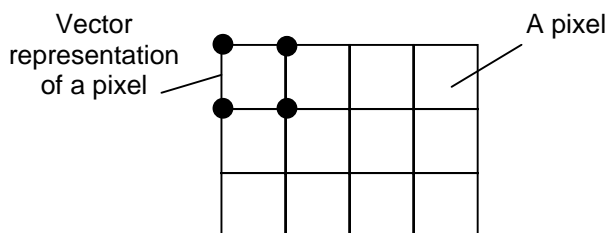


Figure 3: Pixels in Vector Form

3 A CONCEPTUAL MODEL OF SPACE TESSELLATION

Peuquet (1984) have proposed a conceptual model for spatial models. We will address the same problem from a tessellation point of view. According to the definition in a

previous section, a tessellation is a partition of space into non-overlapping cells. We do not make distinction between point-, line-, and area-features. Instead, we regard point features and line features as degenerate cases of area features. This is in conformance with the concept that all geographic features in the real world occupy space, and they are represented as points and lines only due to limitations of scale.

We have identified two broad classes of space tessellation previously, the FPT and SPT. The FPT uses features as the primary units of representation while the SPT uses space cells as the primary units. We will show further that not all space cells are arbitrary but some are constructed under the constraints of features. Take for example a TIN. The triangular patches of a TIN do not represent real world features per se, but they are constructed with reference to spot heights. Another example is the dual of Delaunay triangulation, the Voronoi polygons, which are constructed from point or line features.

We therefore regard SPT to be of two types: arbitrary and constrained by features. The first type produces space cells completely disregarding the features. A well known type of arbitrary SPT produces regular square space cell covering a project area. In fact, all arbitrary SPT used in the digital environment use regular cells. In the analogue environment, however, space cells of random shapes are used.

The second type produces irregular space cells with references to the location and shape of features. There are many examples in this category such as TIN, Voronoi polygons, and quadtrees.

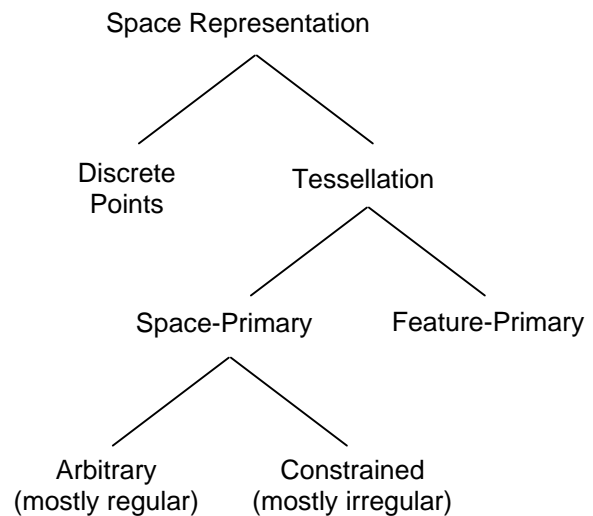


Figure 4: A Conceptual Model of Space Tessellation

4 GEOMETRIC PRIMITIVES FOR IMPLEMENTING SPACE TESSELLATION

Both feature and space cells occupy an area. To physically store these cells in the computer, we need the help of geometric primitives.

There are three types of geometric primitives we normally use. They are points, lines, and pixels. A line is an ordered list of points and a pixel is a point associated with

an area of regular shape. A vector line, as explained earlier, is one that assumes a straight line segment between consecutive points. Connection of points with higher order curves, such as cubic polynomials, are possible although their use in a geographic databases are not very popular.

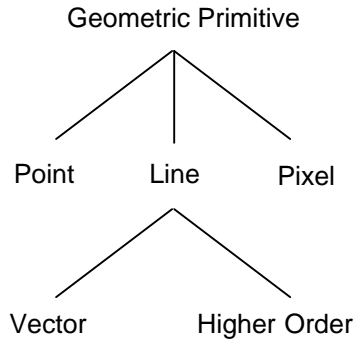


Figure 5: Geometric Primitives

Other than the regular geometric primitives, there are numerous data compressed versions of them. For instance, relative co-ordinates and chain encoding are compression methods for lines. Alternatively, run-length encoding, quadtree encoding, and block encoding are compression method for a matrix of pixels.

It should be noted that we make a distinction between a regular, arbitrary SPT using square cells and a matrix of pixels although they look alike. A tessellation is just a way we interpret space while the pixels are used to physically represent that space. As we have said earlier, it is theoretically possible (although not practically advisable) to represent such a tessellation using a collection of square polygons.

Quadtree also serve dual purposes – it can be either a tessellation method or a compression method depending on how it is being used. For instance, if we use quadtree to create cells for spatial indexing, we are creating a constrained SPT. It is constrained because the partition of the quadtree cells is affected by the distribution of features. When quadtrees are being used to compress the storage of the geometric primitives, the quadtree must be associated with an encoding method such as linear encoding.

5 CONCLUSIONS

In this paper, we have described the various forms of space tessellation and the different geometric primitives that could be used to implement them. At the same time, we have proposed a taxonomy for a more systematic description to avoid the confusion created by the use of the terms “vector” and “raster”. In our proposal, we differentiate between a conceptual model and an implementation model for tessellation. The conceptual model describes how users perceive the partition of space into manageable units at an abstract level.

We recognise here two tessellations – feature-primary and space-primary. The judgement on whether a tessellation is feature-primary or space-primary depends on whether a feature or a portion of space not directly related to features is used as the main storage unit. For

space-primary tessellation, we also recognise that the space cells resulting could be formed arbitrarily or constrained by the presence of features. Arbitrary space cell are mostly regular in shape while the constrained ones are mostly irregular in shape.

We suggest that there are three geometric primitives that could be used to implement the feature or space cells. They are points, lines, and pixels. Both lines and pixels are based on points in that a line is a series of ordered points and a pixel is a point associated with a square or rectangular area. The geometric primitives can be compressed to save storage, and there are numerous compression techniques possible.

As a result, what we traditionally called a vector format is a feature-primary tessellation implemented with vector lines and a raster format is an arbitrary space-primary tessellation implemented with pixels. This taxonomy allows the flexibility of describing some of the non-traditional formats, such as the use of pixels instead of vector lines to represent an area feature.

6 REFERENCES

- Peuquet, D. J., 1981. “An examination of techniques for reformatting digital cartographic data/ Part I: The raster-to-vector process.” *Cartographica*, 18:34-48.
- Peuquet, D. J., 1984. “A conceptual framework and comparison of spatial models.” *Cartographica*, 21: 66 - 113.