23

Accessing outline shape information efficiently within a large database II : database compaction techniques

Peter L. Main
British Museum Research Laboratory

23.1 Introduction

This paper reports on work following on from ideas presented in Main 1986, where a structure was proposed for large databases containing outline shape information from archaeological artefacts. Setting up the database involves applying a clustering algorithm to group 'similar' outlines together, and merging these groups progressively to form a tree structure (or a network structure if the chosen algorithm allows overlapping clusters). The database can then be searched or browsed by moving around those branches of the tree most 'relevant' to the user. An important consideration in the database design is that complete outlines should be available at any point in the search for display on a graphics VDU. The form in which outlines are stored and compared is a modified form of the tangent profile, whose generation from digitised outline data is described in detail in Main 1981, and the modified form in Main 1986. Sections 23.2 and 23.3 below address two particular aspects of the database design:

1. techniques for reducing the size of the tangent profile records without losing information; and
2. ways of defining links between adjacent levels of the tree.

Both topics have considerable implications for the overall size of the database.

23.2 Non-uniform sampling of tangent profiles

The storage format for outline shapes proposed in Main 1986 was that of the sampled tangent profile (STP), where each outline is represented as S (say) tangent angle values sampled from the outline's tangent profile (TP). The TP is a function of tangent angle versus arc-length measured from a reference point on the outline and is scaled to have total arc-length = 1.0., thus removing the effect of size. This function is then sampled at equal intervals of arc-length to give S sampled values of tangent angle, and we will refer to this as a uniform STP of density S. If all outlines in a database are sampled at the same density we can store the STPs in a randomly accessed, fixed record length file which we will call the STP file.

This simple sampling scheme allows fast record access, fast comparison of STPs using microcoded vector arithmetic routines, and easy averaging of groups of STPs to form group

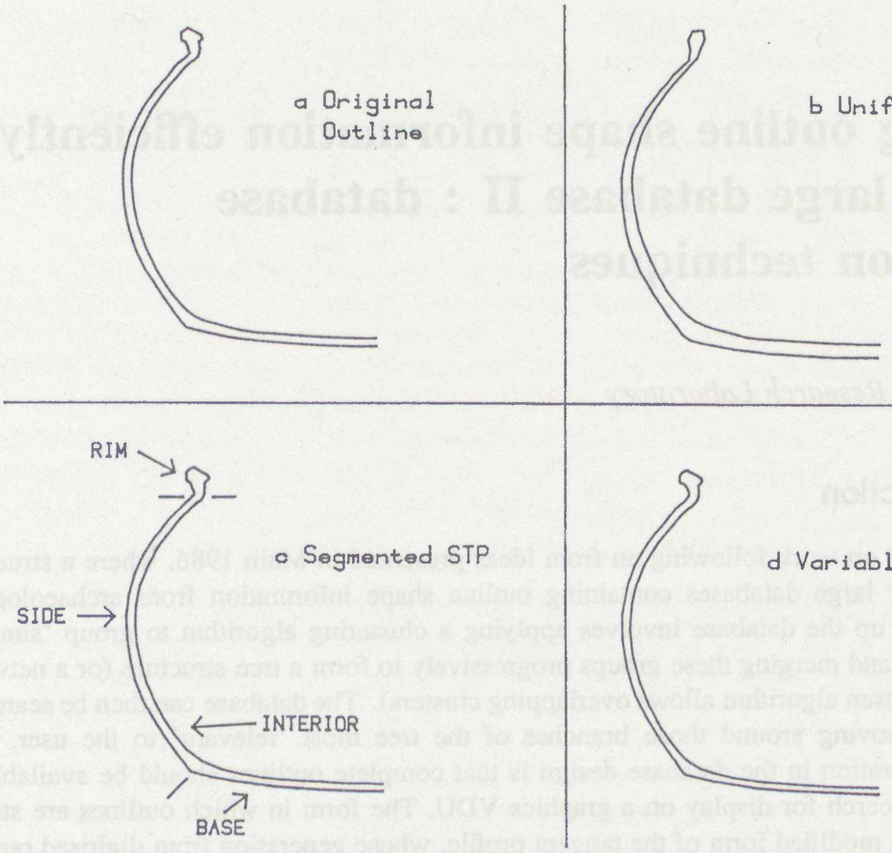

Fig. 23.1: Original 'exact' medieval pot outline generated from an unsampled tangent profile, and three outlines generated from STPs sampled at density 100 in various ways.

centroids. The major drawback of uniform sampling, however, is precisely that it *is* uniform. This is inappropriate on artefact outlines that typically have a high variability in curvature. Consider, for example, the medieval cooking-pot outline in Fig. 23.1a, where the curvature variation is much greater in the rim area than in the base or side of the pot. The detail of the rim shape needs to be preserved, not just for good quality display but also for comparative purposes since rims are often highly diagnostic features in medieval pottery typology. Provided that enough detail from the rim has been digitised, the resulting TP will reflect that level of detail, but to retain the information in a uniform STP would require a sampling density that is far higher than is necessary for the rest of the pot's outline. This results in the database as a whole being much larger than necessary, and also the speed of search will be degraded due to the long vector comparisons between STPs.

We will now look at two approaches to solving this problem by sampling the TP non-uniformly. Whatever sampling scheme we use, it remains important that it is the same for all outlines in the database, otherwise the advantages of fixed record length and easy comparison of STPs are lost.

23.2.1 Segmented STPs

Segmenting tangent profiles relies on being able to divide each outline into regions (corresponding to intervals of arc-length) within which curvature variability is relatively constant. For example, a cooking-pot outline could be segmented into base, side, rim, and interior. Each segment can then be sampled uniformly at a density proportional to its mean curvature variation. Since we require the sampling scheme to be the same for all outlines, the density chosen for each segment should be based on the mean curvature variation figure for that segment averaged over the whole database.

Fig. 23.1 shows how dramatically segmentation can improve the information content of the STP for a given overall sampling density. Fig. 23.1a shows a section through a medieval pot which has been generated from the unsampled TP, and can therefore be regarded as 'exact'. Fig. 23.1b shows an outline of the same artefact generated from a uniform STP of density 100, and Fig. 23.1c shows the outline generated from an STP segmented into base, side, rim and interior, and having segment densities 10, 15, 50 and 25 respectively. Note that the overall densities in Fig. 23.1b and Fig. 23.1c are the same.

The following points can be made about the use of segmentation in sampling TPs.

1. The saving in record length resulting from the use of segmentation will be offset slightly by the fact that in order to display a segmented STP correctly in Cartesian form we need to know the relative lengths of the segments, which will vary from one artefact to another. These lengths therefore need to be stored along with the STP itself as part of each STP file record, and averaged along with the STP as group centroids are formed.

2. In practice, segmenting outlines seems most feasible where the segments can be easily identified at the digitising stage and can be chosen to coincide with identifiable 'features' of the artefact (e.g. base, side, rim of a pot). The break-points between segments therefore need to be identifiable on all outlines to be stored in the database. With some types of artefact this can be a problem—either the extent of a feature may be difficult to define (where does the rim of a pot begin and end?) or the break-point may not exist at all (how much of a round-bottomed pot is the base?).

3. It is important to realise that comparison between two STPs has now undergone a qualitative change. We are now comparing segment with segment whereas with a non-segmented STP we could well be comparing, for example, part of the base of one pot with part of the side of another if the relative base lengths were significantly different. In one sense this seems an improvement since we are now comparing like with like, but on the other hand we have lost an important aspect of overall shape discrimination, since pots with narrow bases and tall sides could appear very similar to those with wide bases and short sides. We need, therefore, to include a component in the distance measure that takes account of the difference in the lengths of corresponding segments between the outlines being compared. This can be easily done, of course, since segment length information has been stored in the STP file records.

4. Where segment boundaries typically occur at a sharn corner in the outline (*e.g.* the base/side boundary of a pot), segmented STPs have the advantage that they preserve the corner when displayed in Cartesian form, because the start and end points of each segment

are always sampled. Group centroid display in particular is very significantly improved since, when uniform STP centroids are displayed there is always a smoothing effect at corners. This is because the corner actually occurs at slightly different positions along the outline in the individual group members, and this ‘rolls out’ the corner of the centroid.

Thus, although segmented STPs do have a number of advantages over uniform STPs, their use relies on being able to segment all outlines unambiguously and in a consistent way over the whole database, and for some classes of artefact this may be difficult or impossible. We turn now to another form of non-uniform sampling that does not require the outlines to be segmented.

23.2.2 Variable STPs

Variable STPs involve sampling the TP at variable intervals along its length. As with segmented STPs, our aim is to sample the TP most densely where its curvature variability is highest. We need, therefore, to measure curvature variability as a function of arc-length in order to generate a sampling scheme for the TPs. We require that this scheme is the same for all outlines in the database, so the curvature variability function needs to be made representative of the whole database in some way. This has been achieved by the following procedure. Assume that we are looking for a variable sampling scheme of overall density, D.

1. Sample each TP uniformly at high density (3 D, say). Do not write the STPs to disc, but accumulate in memory the overall centroid (‘grand mean’) of all the STPs.
2. Take the second difference of this centroid STP. The first difference gives a profile of curvature, and the second difference gives a profile of curvature variation. Take the modulus of this function. We now have a function of arc length whose peaks correspond to regions of high curvature variation in the artefact outlines. It broadly represents the whole database, since it is derived from the overall centroid.
3. Use this function to derive a sampling profile having overall density D and having the property that regions of high sampling density correspond to peaks of the function. A sampling function and the derived sampling profile of density 60 are shown in Fig. 23.2. They are derived from the medieval pot of Fig. 23.1a.
4. Re-sample all the TPs using the sampling profile and write the STPs to the STP file. The sampling profile itself is also written as an extra record to the STP file since it will be needed to display the STPs correctly as Cartesian outlines, and for similarity calculations.

Fig. 23.1d shows once more the Cartesian outline of a medieval cooking-pot, but this time displayed from a variable STP of density 100. This can be compared with Fig. 23.1b, which shows the same artefact displayed from a uniform STP of the same density.

No extra information needs to be added to variable STP records as is necessary with segmented STPs. Although we need to store the sampling profile itself, this occupies only one STP file record. The main overhead in using variable STPs is a slight degradation in the speed of comparison between STPs. The standard distance measure between uniform STPs is a simple sum of absolute differences between the sampled values, whereas with variable STPs the equivalent measure becomes a weighted sum—that is, one extra vector multiplication is required. For a given accuracy of outline, however, this is offset by the fact that the vectors involved will have been reduced in length.

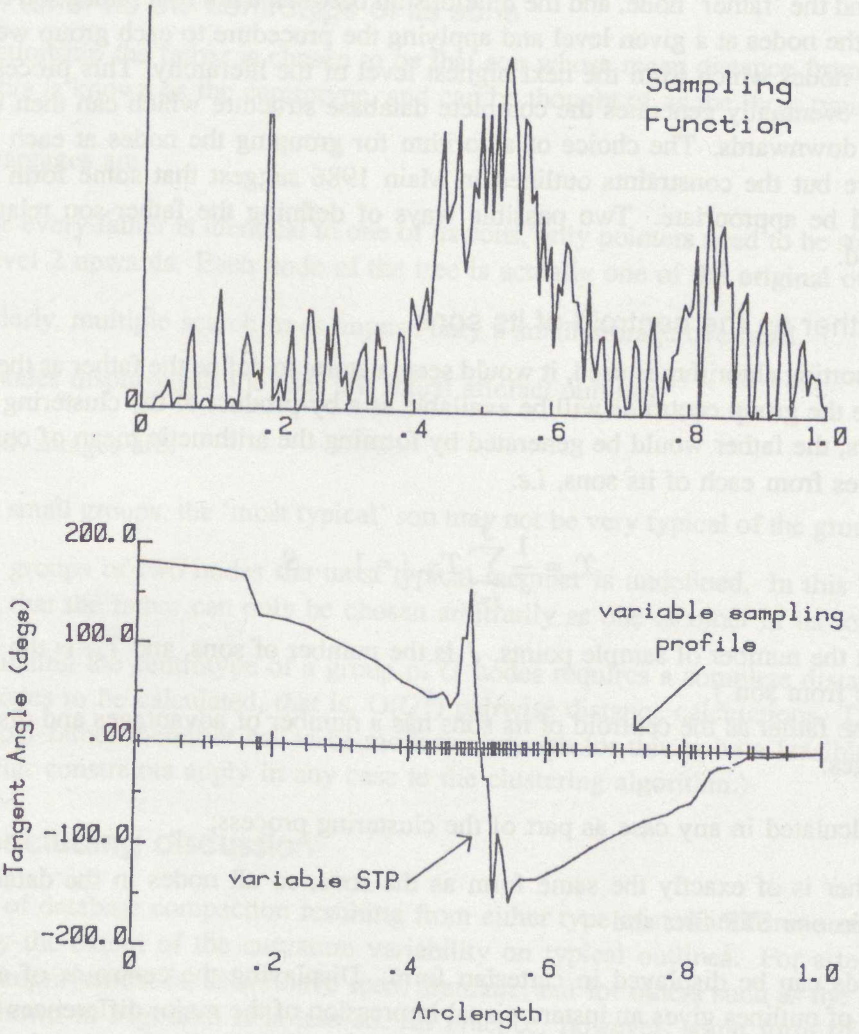

Fig. 23.2: (Above) The sampling function generated from the pot outline in Fig. 23.1a. (Below) The derived variable STP and sampling profile.

23.3 Definition of father-son relationships

In generating a hierarchical database structure such as we are proposing here, there is an immediate requirement that we define a procedure for generating from a given group of nodes (each node representing an outline) at level L (say) on the tree, a single node at level L+1 that is in some sense representative of the group. These are commonly referred to as, respectively, the 'son' nodes and the 'father' node, and the relationship between them as a father-son relationship. By grouping the nodes at a given level and applying the procedure to each group we generate a new series of nodes which form the next highest level of the hierarchy. This process, repeated at each level, eventually generates the complete database structure which can then be searched from the top downwards. The choice of algorithm for grouping the nodes at each level is not discussed here but the constraints outlined in Main 1986 suggest that some form of centroid sorting would be appropriate. Two possible ways of defining the father-son relationship are now described.

23.3.1 Father as the centroid of its sons

If a centroid sorting algorithm is used, it would seem natural to define the father as the centroid of its sons, since the group centroids will be available as a by-product of the clustering process. In terms of STPs, the father would be generated by forming the arithmetic mean of corresponding sampled values from each of its sons, *i.e.*

$$T_i = \frac{1}{J}\sum_{j=1}^{J} T_{ij},\, i = 1, \ldots, S$$

where S is the number of sample points, J is the number of sons, and T_{ij} is the ith sampled tangent angle from son j.

Defining the father as the centroid of its sons has a number of advantages and disadvantages. The advantages:

1. it is calculated in any case as part of the clustering process;
2. the father is of exactly the same form as the sons, so all nodes in the database can be stored in one STP file; and
3. centroids can be displayed in cartesian form. Displaying the centroids of a number of groups of outlines gives an instant visual impression of the major differences between the groups (see Fig. 23.3). This is a most valuable facility in the context of an interactive graphics database.

The disadvantages are as follows:

1. There is a considerable increase in the size of the database as a result of storing a complete STP for each father node. That is, the database is proportional in size to the total number of nodes rather than to the original number of outlines.
2. This overhead becomes worse if more than one tree structure is to be used to search the database. This might be necessary where more than one similarity measure is to be available for searching the database, resulting in different search trees and hence different centroids.

3. Since we wish to display father nodes to the user of the system, we need to take steps to make him aware that they are not outlines of real artefacts.

A second possibility for the father-son relationship is now suggested. This overcomes the disadvantages of the group centroid, but has some of its own.

23.3.2 Father as the centrotype of its sons

In this relationship the father is chosen to be that son whose mean distance from all other sons is least. This is known as the centrotype, and can be thought of as the most typical member of a group.

The advantages are:

1. Since every father is identical to one of its sons, only pointers need to be stored for nodes of level 2 upwards. Each node of the tree is actually one of the original outlines.
2. Similarly, multiple search trees impose only a small storage overhead.
3. All nodes displayed to the user are actual artefact outlines.

The disadvantages are:

1. With small groups, the 'most typical' son may not be very typical of the group as a whole.
2. With groups of two nodes the most typical member is undefined. In this case it would seem that the father can only be chosen arbitrarily as one or other of its sons.
3. Calculating the centrotype of a group of G nodes requires a complete distance matrix of the nodes to be calculated, that is, $O(G^2)$ pairwise distance calculations. The number of sons per father therefore has to be moderate enough for this to be a feasible proposition. (Similar constraints apply in any case to the clustering algorithm.)

23.4 Concluding discussion

The degree of database compaction resulting from either type of non-uniform sampling will be governed by the extent of the curvature variability on typical outlines. For artefacts such as pottery the improvement is, as we have seen, dramatic, but for others such as the Early Bronze Age axes shown in Fig. 23.3 it is less so. In practice, however, some form of non-uniform sampling seems always likely to be worthwhile.

There is nothing to prevent both segmentation and variable sampling being employed simultaneously. We could first segment each outline and then calculate a variable sampling profile for each segment in an analogous manner to that described in section 23.2 above. Unless fast STP comparison is a critical consideration, it is probably always worth using variable rather than uniform sampling. Thus, the real choice we are left with is whether to segment the outlines or not. This decision depends firstly on how feasible this is to do, and secondly on whether the qualitatively different type of shape comparison this gives rise to seems appropriate. The considerations that apply when calculating distance between segmented tangent profiles are discussed more fully in Chapter 6 of Main 1981.

The choice of father-son relationship will of course affect the outcome of searching the database. As a simple example, assume that a tree of N levels is to be searched by comparing

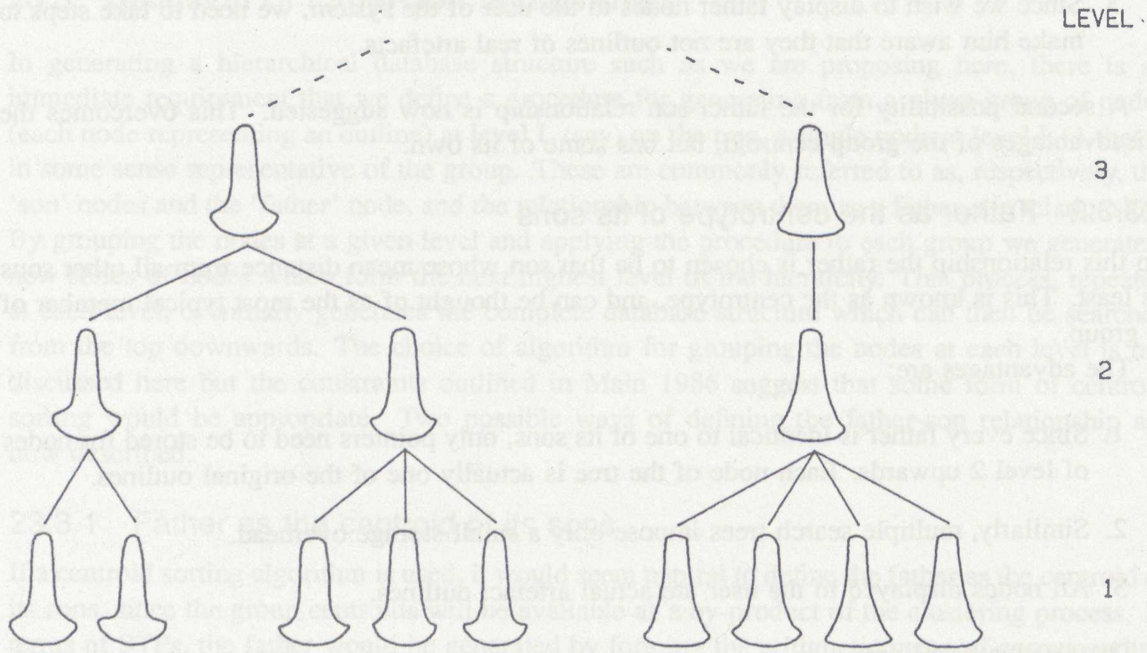

Fig. 23.3: Part of a database of Early Bronze Age axe outlines, which uses group centroid as the father relationship.

a target outline (converted to STP form) with all nodes at the level $N - 1$, passing down that branch closest to the target to a group of level $N - 2$ son nodes, and so on until a group of level 1 nodes is reached. These form the result of the search. If the father-son relationship is that of centroid, this search strategy accords well with the way the database has been structured, since it is the objective of centroid sorting algorithms, roughly speaking, to find compact clusters whose centroids are as well separated as possible. There is, on the other hand, no obvious relationship between the idea of the most typical member of a group and the centroid sorting procedure. However, the centrotype is the item nearest to the centroid for Euclidean distance models (Sneath & Sokal 1973). Although in the case of STPs the distance measure proposed is not Euclidean, experiments on Early Bronze Age axe outlines suggest that the centroid and centrotype are quite close. Although this requires further investigation over a range of artefacts and other distance measures, these findings reassure us that searching a tree with fathers defined as most typical sons is likely to give sensible results, since the process is in fact very similar to what would result from defining fathers to be centroids. Furthermore, the most typical son becomes easier to find when the tree structure is being built. Rather than calculating pairwise distances between all the sons we simply find the son nearest to the centroid, the latter being already available from the clustering algorithm. In other words we require once more only $O(G)$ vector operations rather than $O(G^2)$.

We are left therefore with two conflicting aims. First, we would like to keep the database storage overheads as low as possible by defining all nodes to be actual artefact outlines, that is, by choosing the father as its most typical son. We would also, however, like the father to be as representative of its sons as possible for display purposes, when searching or browsing the database. For groups where the most typical son is not very typical (or, equivalently, not close to the group centroid) we would prefer to store and display the centroid as the father-node. This dichotomy suggests a hybrid solution to the database structure, which could be implemented along the following lines.

1. At each level use a form of centroid sorting to group the nodes.
2. For each group find the group member Gx closest to the centroid.
3. If the distance from Gx to the centroid is greater than some threshold value, or if the group has only two members, choose the father to be the group centroid and add the centroid STP to the STP file. Otherwise choose the father to be Gx. In this case no new record is written to the STP file—the father node is simply pointed to the same STP file record as Gx.

The choice of threshold value needs consideration of course, and will presumably depend on the current level within the tree since moving up the tree reflects progressively coarser shape comparison. Finally, we should note that the procedure outlined above gives rise to two classes of node: those that are outlines of real artefacts, and those that are not. It seems advisable that this distinction is made obvious to the user when nodes are displayed, for example by using different colours.

References

MAIN, P. L. 1981. *A method for the computer storage and comparison of the outline shapes of archaeological artefacts.* PhD thesis, CNAA.

MAIN, P. L. 1986. 'Accessing outline shape information efficiently within a large database', *in Computer Applications in Archaeology 1986*, University of Birmingham.

SNEATH, P. H. A. & R. R. SOKAL 1973. *Numerical Taxonomy,* Freeman, San Francisco.