Pattern Recognition Letters 12 (1991) 537-541
North-Holland

September 1991

Adaptive partitioning strategies for ternary tree structures

Jeffrey K. Uhlmann

Information Technology Division (code 5570), Naval Research Laboratory, Washington, DC 20375, USA

Received 5 February 1991
Revised 15 April 1991

Abstract

Uhlmann, J.K., Adaptive partitioning strategies for ternary tree structures, Pattern Recognition Letters 12 (1991) 537-541.

Test results of three adaptive partition strategies for constructing multidimensional ternary trees are presented. Such trees can be used to efficiently identify correlated measurements with error bounds that can be represented as finite volumes.

Keywords. Measurement correlation, data fusion, computational geometry, multidimensional trees, adaptive decomposition.

Introduction

Given a set of N d-dimensional hyperrectangles, or *boxes*, an important class of problems requires the efficient identification of the subsets that intersect a given query box. The optimal worst-case scaling (i.e., the lowest possible upper-bound scaling) performance for any algorithm requiring only linear storage can be no better than $O(N^{1-1/d}+m)$ [1], where m is the number of actual intersections. In many cases, however, nonuniformly distributed data has an effective search dimensionality somewhat less than d. In other words, distribution characteristics can often be exploited to reduce the worst-case computation time required for identifying intersections. In this paper, various approaches to adaptive decomposition are examined for the exploitation of anisotropic distribution features in static multidimensional tree structures. The results are of particular relevance to areas such as robotic manipulation, molecular dynamics simulations, and tracking and correlation [2-6]. More generally, however, the efficient identification of multidimensional volumetric intersections is a combinatorial issue inherent to a wide variety of pattern recognition and data fusion problems. For example, any measurement process that yields a vector having finite error bounds associated with each element can be viewed as a hyperrectangle. A necessary condition for two measurements to be of the same object, then, is that their measurement boxes intersect. Surprisingly, the above quoted scaling for nonadaptive search structures appears to imply that measurement correlation becomes more difficult (approaches quadratic scaling) as the amount of information, i.e., the dimensionality of

the measurements, increases. This counter-intuitive phenomenon can be mitigated only through the use of adaptive decomposition techniques.

Multidimensional ternary trees

A multidimensional tree is a binary tree in which the set of d-dimensional points is partitioned at each node according to one of the d coordinates [7,8]. The choice of discriminating coordinate is often made by simply cycling through the coordinates in a nonpreferential manner. Thus, a node will contain (either implicitly or explicitly) identification of the discriminating coordinate as well as pointers to the set of points whose values for the discriminating coordinate are greater than that of the median and to the set of points whose values are less than the median. This procedure results in a data representation that is capable of satisfying a wide variety of search queries on points. It must be enhanced, however, to accommodate volumetric objects. For efficiency it is often assumed that the volumes are approximated by isothetic rectangles, i.e., boxes whose sides are parallel to the coordinate axes. However, since the projection of boxes onto an axis results in a set of intervals, possibly overlapping, the notion of a projection median that can always divide a dataset into two disjoint sets can no longer be applied. Thus, a multidimensional tree that is enhanced to handle volume queries may not be strictly binary.

The multidimensional ternary tree search is a ternary tree in which the set of d-dimensional boxes is partitioned at each node according to one of the d coordinates. The resulting tree is ternary because at each node the set is partitioned into a set of boxes that lie entirely to the left of the partitioning plane, a set of boxes that lie entirely to the right of the partitioning plane, and a set of boxes that are intersected by the partitioning plane [9]. The method for searching the tree then simply involves the determination of which set the search box would be assigned according to the partitioning plane at each node. For example, if the search box lies entirely to the left of the partitioning plane, then the subtree of boxes that lie entirely to the right of it can be ignored. Similarly, if the search box lies entirely to the right of the partitioning plane, then the subtree of boxes that lie entirely to the left of it can be ignored. In the case where the partitioning plane intersects the search box, all subtrees must be examined.

(The levels at which the search box intersects the partitioning plane contribute significantly to the overall cost of the search. This is because at least three paths beginning at the node associated with the plane must be examined to the full depth of the tree in order to determine if there are any more boxes in the neighborhood. Even if no plane intersects the neighborhood, two complete paths from the root to terminal nodes must still be examined. A solution to this source of inefficiency is to construct the tree structure so that each of its nodes represents *two* partitioning hyperplanes rather than one [8]. The second partitioning plane is determined by the left side of the nearest box on the right of the initial partitioning plane. These two planes delimit an interval for which only the middle subtree must be examined. This enhancement is particularly advantageous when the projection density of the items in the structure is low or when the query ranges are known to be correlated with data in the search structure. The latter occurs because such correlations increase the likelihood that a query box will straddle a partitioning plane.)

Distribution dependencies

Distribution considerations arise in a number of contexts. A common situation is to have a set of k-coordinate boxes embedded in an isothetic subspace of dimensionality less than k. In other words, the vertices of the boxes have approximately equal values in one or more of the coordinates. This can substantially reduce the efficiency of the search structure because many or all of the boxes may intersect the partitioning planes associated with those coordinates. In order to minimize this effect, it is necessary to use distribution information to select the partitioning planes at each internal node of the tree. It can be shown that this type of adaptive decomposition strategy not only mitigates against simple anisotropy, but also against the effects of clustering.

In principle, the selection process should utilize information about the expected distribution of query boxes as well as the distribution of boxes in the search structure. In general, however, query information cannot be assumed. Thus, the objective of the selection strategy should be to optimize those performance variables for which complete information is available. The most important such variable is the average size of the subtrees resulting from partition intersections. This variable is critical to search efficiency because it represents the amount of imbalance in the tree and the amount of obligatory search effort associated with each node. Recognizing this fact, the problem then is to identify a selection strategy that optimizes the tradeoff between improved search performance and increased setup time for the search structure. Clearly, if the number of queries greatly exceeds the number of objects in the search structure, the setup time may not be important. However, many problems require the identification of correlations (signified by volumetric intersections) between datasets of roughly equal size. As a result, tests of various strategies will consider only the case in which the number of queries equals the number of objects in the search structure and the two sets of boxes are strongly correlated in position and extent. Application of the test results to other cases should be relatively straightforward.

Selection strategies

Techniques for selecting partitions in search structures containing point objects [10] are generally inadequate for the volumetric case because the statistical properties they exploit are difficult to apply meaningfully to extended objects. In the volumetric case the subtrees of objects cut by parititions may have a significant impact on the search cost, but this factor is not an issue for simple range queries. Consequently, the adaptive strategies compared below explicitly use extent information. The strategies considered are:

1. Cyclical method. This nonadaptive approach is the most commonly used in multidimensional tree structures. It simply involves cycling through the coordinates in an unbiased fashion. For example, the splitting coordinate for a node at level p could be computed by the formula $p \bmod d$, where d is the dimensionality of the space. The appeal of this approach is that it is easy to implement and incurs virtually no overhead. Its inclusion in these tests, however, is to serve as a baseline for comparison.

2. Extremal method. This approach selects the coordinate upon which the projected data spans the largest interval. This approach is attractive because it incurs very little computational overhead and is effective on uniform distributions within general (i.e., rectangular) isothetic regions.

3. Minimum-density method. This method selects the coordinate upon which the projected data has minimum density. This determination is made by calculating for each coordinate the ratio of the sum of the lengths of the projected box intervals and the length of the spanned interval. The advantage of this approach over the extremal method is that it utilizes information about the distribution of box extents in each coordinate.

4. Median-intersection method. This is the superior method for minimizing the number of partition intersections. It simply selects the coordinate having the fewest intersections with the median. Its disadvantage, however, is that it incurs substantial computational overhead, especially when linear worst-case scaling is demanded for the median calculation.

In order to compare the approaches, tests were performed with varying assumptions about the number of boxes and the distribution of the boxes in the search structure. Although machine dependencies must always be considered, the following results should provide a good indication of relative performance.

Test results

The following Tables 1–4 provide the total processing time in seconds required for n queries, where n is the size of the dataset, for the following

Table 1
Cyclical method

Dataset size	Anisotropic	Cluster	$1/r^2$	Uniform
1 K	0.81	0.72	0.58	0.74
2 K	2.32	2.05	1.68	2.16
4 K	6.09	5.85	4.60	5.64
8 K	18.61	15.73	13.89	14.16
16 K	42.63	37.08	32.90	39.75
32 K	108.04	103.96	72.68	95.99
64 K	245.60	227.51	197.95	225.00
128 K	567.37	550.21	406.86	552.91

Table 3
Minimum-density method

Dataset size	Anisotropic	Cluster	$1/r^2$	Uniform
1 K	0.60	0.60	0.44	0.61
2 K	1.72	1.64	1.24	1.72
4 K	4.33	4.38	3.18	4.45
8 K	12.58	12.26	8.96	11.05
16 K	27.65	27.58	21.26	29.55
32 K	69.10	70.98	44.48	66.36
64 K	161.16	157.27	116.08	158.73
128 K	375.22	368.04	238.39	369.78

distributions of uniformly-sized boxes in 3-dimensions:

1. *Anisotropic* – the distribution of the boxes projected onto each coordinate is uniform, but the density of the projection onto coordinate k is twice that of the projection onto coordinate $k-1$.
2. *Clustered* – clusters of $n/8$ boxes are uniformly distributed.
3. *Inverse-square* – boxes are distributed in a single octant such that the density diminishes as the inverse square of the distance from the origin.
4. *Uniform* – boxes are distributed uniformly.

The volume of the boxes in each search structure was selected so as to assure an average of five intersections per query. In the case of the inverse-square distribution, holding this variable constant implied that a few query boxes had many intersections while most had only one (itself). This accounts for the generally reduced query times for the inverse-square tests. Local high-density regions similarly affect average query times in the cluster tests. Times quoted are averages of 5–100 tests on a Sun4 workstation.

Table 1 presents the results of tests using the cyclical partition method. The relatively small differences in processing times between tests having different distribution assumptions is attributable to the fact that the set of query boxes has the same distribution as the set of boxes in the search structure (since they are the same set in these tests). The fact that distribution information exists that can be exploited for improved query times is revealed by Table 2 of results of tests using the extremal partition strategy.

Table 2 shows an average 35% improvement in query times over the nonadaptive cyclical method. For the only nonsymmetric distribution, the inverse square case, an improvement of more than 40% is gained. These performance gains are also achieved using the minimum-density strategy. See Table 3.

A comparison of Tables 2 and 3 reveals that the performance of the extremal method and the

Table 2
Extremal method

Dataset size	Anisotropic	Cluster	$1/r^2$	Uniform
1 K	0.59	0.60	0.44	0.61
2 K	1.74	1.64	1.22	1.72
4 K	4.33	4.39	3.18	4.43
8 K	12.65	12.27	8.96	11.09
16 K	27.78	27.65	21.32	29.59
32 K	69.01	70.88	44.45	67.51
64 K	161.21	157.34	115.48	158.75
128 K	375.09	368.00	239.59	368.87

Table 4
Median-intersection method

Dataset size	Anisotropic	Cluster	$1/r^2$	Uniform
1 K	0.57	0.57	0.44	0.58
2 K	1.55	1.56	1.12	1.55
4 K	3.97	4.05	2.98	4.06
8 K	10.49	10.54	8.60	11.59
16 K	29.34	28.21	20.38	25.95
32 K	66.78	63.01	48.45	62.16
64 K	153.15	147.61	107.37	150.03
128 K	357.71	349.13	226.32	351.25

minimum-density method are virtually identical. This suggests that the two strategies tend to select the same partitions. Because the extremal method incurs significantly less computational overhead during construction of the search structures, it is probably preferable to the minimum-density method. Table 4, though, shows that the median-intersection method is clearly superior in terms of query-time performance.

Table 4 reveals an average 5% improvement over the previous two strategies tested and almost a 40% average improvement over the cyclical selection method. When the processing time for construction of the search structures is considered, however, the total time for processing the set of queries using the median-intersection method is roughly comparable to that of the extremal method.

Discussion

Test results of three adaptive partition strategies have been presented which demonstrate that distribution information can be exploited for improved query-time performance even when the set of query boxes are strongly correlated with the set of boxes in the search structure. This kind of problem arises in a variety of practical applications in which multiple measurements of the same set of objects must be correlated for data fusion [11, 12]. Tests show that the median-intersection strategy for adaptively constructing a ternary tree search structure improves query time by nearly 40% over the cyclical partitioning commonly used for constructing multidimensional search structures.

References

[1] Mehlhorn, K. (1984). *Multidimensional Searching and Computational Geometry*. Springer, Berlin.

[2] Zuniga, M.R., Picone and J.K. Uhlmann (1990). Efficient algorithm for improved gating combinatorics in multiple-target tracking. Submitted to *IEEE Trans. Aerospace Elec. Syst.*, April 1990 (and published as NRL Memo Report 6691).

[3] Uhlmann, J.K. and M.R. Zuniga (1991). Results of an efficient gating algorithmn for large-scale tracking scenarios. *Naval Research Reviews* 1.

[4] Collins, J.B. and J.K. Uhlmann (1990). REAL approach to tracking and correlation for large-scale scenarios. *NRL Review*.

[5] Uhlmann, J.K. and M.R. Zuniga (1991). New approaches to multiple measurement correlation for real-time tracking systems. *NRL Review* (in press).

[6] Uhlmann, J.K., M.R. Zuniga and Picone (1990). Efficient approaches for report/cluster correlation in multiple-target tracking systems. NRL Report 9281.

[7] Bentley, J.L. (1975). Multidimensional binary trees for associative searching. *Comm.* 18(9), 509–517.

[8] Samet, H. (1990). *The Design and Analysis of Spatial Data Structures*. Addison-Wesley, Reading, MA.

[9] Uhlmann, J.K. (1990). Enhancing multidimensioinal tree structures by using a bi-linear decomposition. NRL Report 9282.

[10] Friedman, J.L. Bentley and Finkel (1977). An algorithm for finding best matches in logarithmic expected time. *ACM Trans. Math. Software* 3(3), 209–226.

[11] Collins, J.B. and J.K. Uhlmann (1990). Efficient gating in data association for multivariate Gaussian distributions. Submitted to *IEEE Trans. Aerospace Elec. Syst.*, Feb. 1990.

[12] Collins, J.B. and J.K. Uhlmann (1991). An efficient general purpose correlation algorithm for multidimensional data fusion. NRL Report.