A SPATIAL DECOMPOSITION BASED APPROACH FOR MATCHING POINT COLLECTIONS

Dr. Marius Dorian ZAHARIA [1,2]
Dr. Jan-Mark GEUSEBROEK [3]

[1] *Associate Professor, Computer Science and Engineering Department*
University POLITEHNICA of Bucharest
E-mail: zaharia@cs.pub.ro
[3] *Intelligent Sensory Information Systems Research Group,*
University of Amsterdam
E-mail: mark@science.uva.nl

Abstract: The paper describes a method to identify (label) the corresponding points of two 2D-point collections. This technique is based on triangulating the point sets and the usage of a fuzzy procedure to test the isomorphism of the PM-trees associated to the above mentioned triangulations. An analysis of the robustness of the method with respect to small random perturbations of the point positions is also presented.
Keywords: Delaunay triangulation, polygonal map tree, tree isomorphism, threshold based classification, structural pattern recognition.

1. Introduction

The problem of matching point collections is encountered in various fields of application as computer vision, robotics, information retrieval in image databases and artificial intelligence. Different approaches suitable to solve this problem have been presented in the pattern recognition literature (Luo and Hancock, 1999).

As an example, in the case of intelligent autonomous robots low level image processing algorithms establish characteristic points of the sampled images captured by the robot vision subsystem; these points describe the environment in which the robot operates. In this above mentioned vision system the environment is static and the point matching technique could assist the robot to correctly recognize the environment objects and consequently correctly determine its position relatively to them. Another direct application of the method described in the present paper is the search of a given image in an image database. Some experimental results concerning this type of application are presented in the third section of this paper.

[2] The author is actually postdoctoral researcher at the University of Amsterdam, The Netherlands

2. Description of the method

The method proposed is a structural pattern recognition technique. The intrinsic structure of the point collection is captured by finding its Delaunay triangulation and decomposing the space of this triangulation using a PM-tree. The Delaunay triangulation of a set of points $S=\{P_1, P_2, \ldots, P_n\}$ has the remarkable property that the triangles that compose it tend to equiangularity (Laszlo, 1996). If $T=\Delta P_iP_jP_k$ is an arbitrary triangle of the Delaunay triangulation of S then its circumcircle does not contain any other point of S. The possibility of slivers occurrence is reduced because, in case of a Delaunay triangulation, slivers have usually at last one side from the convex hull of the data set. If the point collection does not contain any 4 cocircular points, the Delaunay triangulation is unique. This property makes the Delaunay triangulation suitable to represent the neighborhood relationship between the points of a given data set.

In order to correctly determine the matching points of S_1 and S_2, these two sets have to be properly aligned. This is achieved by identifying the points P_{m1} and P_{m2} of the two collections that have the distances to the centroids (C_1 and C_2) of the two sets maximal. This method is easily implemented and worked correctly on all the tested data sets A more accurate alignment technique is based on the eigenvalue analysis (Gonzalez and Woods, 2002) of the covariance matrices that characterize the point collections. In the alignment phase of the method, the points from S_1 and S_2 support:

- rotations that make C_1P_{m1} and C_2P_{m2} aligned to the x axis of the data space reference frame,
- translations that move C_1 and C_2 in a fixed position and
- uniform scaling that frame the convex hull of every point set in the rectangle $[0,1]\times[0,1]$.

This practically "normalizes" the two point sets.
Following this preliminary alignment step the Delaunay triangulations of the two point sets are determined.

The adjacency properties characteristic to the data points are quantitatively captured using a spatial decomposition technique suitable to describe polygonal mesh data. It is the PM-quadtree invented by Samet, Nelson and Weber (Nelson and Samet, 1986). This hierarchical structure represents, without loss of precision, polygonal mesh data and is robust with respect to rigid body transformations applied to the geometric data. From the many variants described in the specialized literature (Samet, 1990) was chosen the one that implements the following decomposition rules:

1. At most one vertex may lie in a region corresponding to a PM-tree leaf node.
2. If one spatial decomposition region (R) contains one vertex (V) then it is not allowed that edges not containing (V) intersect (R).
3. If a region (R) of the PM-tree does not contain any vertex, it may contain only parts of edges that have one common vertex exterior to (R).
4. The regions corresponding to the PM-tree leaf nodes are maximal.

Figure 1 shows the PM spatial decomposition corresponding to a Delaunay triangulation of a 6 points data set and its associated tree.

The identification and labeling of the matching points from the two collections (whose Delaunay triangulations are modeled by the two PM-trees) are performed using an algorithm based on a tree-isomorphism technique. Small perturbations of the point positions could induce large variations of the corresponding tree heights. Generally the tree height increases if two points of the set, connected by a triangulation edge are placed very nearly one to another. This corresponds to slivers in the triangulation, case that we have seen, occurs seldom in the case of Delaunay triangulations. Consequently the tree isomorphism test will take into consideration only the tree nodes placed at levels less than the minimal height of the two trees. The geometric information referring to the vertices of the Delaunay triangulation is kept in the tree leaves, the topological information is codified by the tree shape. The procedure verifying the similarity of the point collections identifies the leaves (situated on the same level) containing correspondent vertices and associates to each pair of such leaves a weight proportional to their level. Finally the procedure computes the percentage of leaves that contained correspondent points relatively to the total number of leaves containing data points. The point collections are considered similar if this percentage is greater than a predefined threshold.

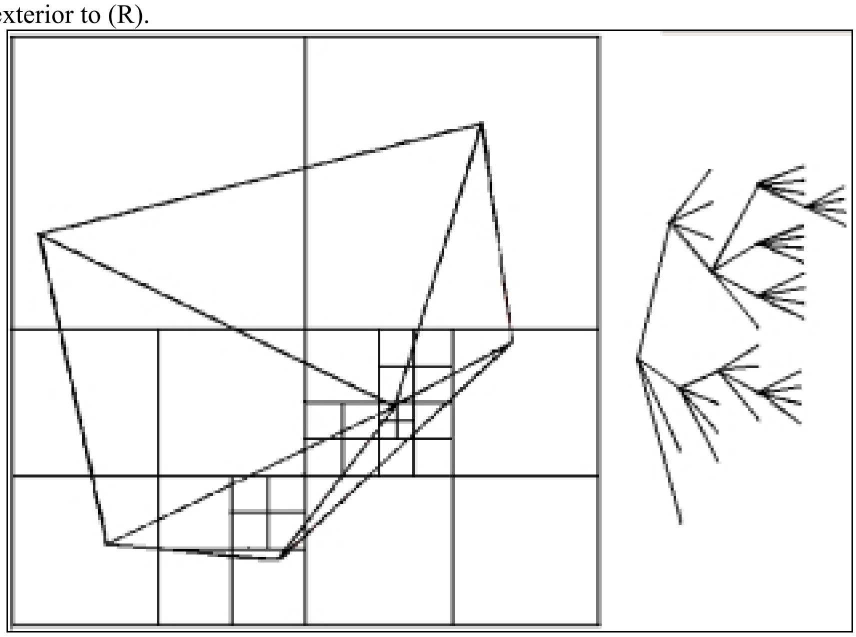

Figure 1 – The spatial decomposition characteristic to a PM-tree

3. Implementation issues

The storage structure of a PM-tree is represented by:

```
enum {FRUNZA, INTERN};
typedef struct pmt {
  rpunct c;
//center of the spatial region
//associated to the current tree node
  double lg;
//length of the side of the spatial (square) region
  lista_muchii lm;
//The list of edges crossing the spatial region  and
//satisfying the PM-tree validity criterion
  int tipnod;
//node type (internal or leaf)
  struct pmt *fiu[4];
//references to the sons of the current node
} *arbore_PM;
```

The skeleton of the fuzzy PM-tree isomorphism procedure (that is similar to the procedure labeling the matching data points) is given by:

```
void izo_PM2(int k, arbore_PM a, arbore_PM b)
{ int i;
  if(k>NLEV) return;
  if(a==NULL) {
    sumafrizo+=k*(b==NULL);
    return;} /*1*/
  if(b==NULL) {
    sumafrizo+=k*(a==NULL);
    return;} /*2*/
  if(a->tipnod==LEAF &&
     b->tipnod==INTERNAL){ …
    return;
  } /*3*/
  if(a->tipnod==INTERNAL &&
     b->tipnod==LEAF){ … return; } /*4*/
  if(a->tipnod==LEAF &&
     b->tipnod==LEAF) { /*5*/
    sumafrizo+=k;
    return;
  }
  for(i=0; i<4; i++){

izo_PM2(k+1, a->fiu[i],
           b->fiu[i]);}
 return;
 }
```

In case of test /*3*/ the labeling procedure identifies the vertex V of the polygonal mesh from the spatial region associated with node a (if such a vertex exists) and tries to find V in one of the edge lists associated with the leaves of the tree dominated by node b. If V is found it will be labeled. Test /*4*/ requires a similar treatment.

From the asymptotic analysis point of view, the phases of the algorithm have the following behavior: point alignment $O(n\log_2(n))$ (due to convex hull determination), building of the PM-tree $O(n\log_2(n))$, the process of testing the tree isomorphism is linear in the number of tree nodes. If this method is used for information retrieval in an image database, the associated PM-trees of the characteristic points of the images in the database can be computed in a preprocessing stage of the application. The query processing requires only an isomorphism test between the trees associated with the query image and these associated to the images from the database.

Let S_1 and S_2 be two point collections. The set S_2 results from applying a Euclidean 2D transformation and a random perturbation to each point of S_1. Table 1 presents the number of the data points the procedure was able to label. One could observe that the results of the matching procedure are strongly dependent on the amplitude of the perturbation. The results were satisfactory for data sets having some hundreds (thousands) elements. The main computational effort was done for building the PM-tree, here the results could be improved considering that in the tests was used an $O(n^2)$ algorithm and this building procedure is not optimal. The labeling algorithm is not time consuming and in the applications where the building of the PM-tree could be done in a preprocessing phase, the matching procedure is reliable and efficient. In Table 1, null times are figured if the reported time was less than the intrusion specific to the used timing measurement mechanism.

The application was implemented in Visual C++ under the .NET framework. The tests were done on a Intel Pentium IV processor operating at 1.6 GHz. The times are specified in milliseconds. The data space is the unit rectangle and consequently a perturbation of 0.01 is equal to 1% of the total length of the data space.

Table 1 Implementation results

No. points	No. nodes	H-Tree	PM time	Perturbation Amplitude	Label time	No. labels	Lab. %
25	261	9	20	0			
25	253	9	20	0.005	0	24	96
25	285	13	20	0.005	30	25	100
50	669	8	40	0			
50	641	9	40	0.005	0	49	98
50	937	12	70	0.01	0	44	88
50	721	10	80	0.01	0	26	52
100	1773	13	250	0			
100	1841	13	350	0.005	20	68	68
100	1225	11	330	0.01	30	46	46
500	5913	16	3284	0			
500	6405	14	5377	0.005	110	257	51.4
500	5425	13	6659	0.01	140	265	53
1000	11745	15	20900	0			
1000	10097	13	28090	0.005	741	669	67
1000	11521	17	37150	0.01	891	644	64

4. Conclusion

The point matching method presented above is suitable to be used for searching in image databases and in computer vision applications. In case of high point densities the method is sensitive to small perturbations of the point positions. The main computational effort is necessary for the PM-tree construction. If this process could be accomplished in a preprocessing stage of the application the matching could be done in real-time.

References

Gonzalez, R. C. and R. E. Woods (2002) *Digital Image Processing*, Prentice Hall Inc., Upper Saddle River, New Jersey 07458, ISBN 0-13-094650-8.

Laszlo, M. J. (1996) *Computational Geometry and Computer Graphics in C++*, Prentice Hall Inc., Upper Saddle River NJ 07458, ISBN 0-13-290842-5

Luo, B. and E. R. Hancock (1999) Matching point sets using Procrustes alignment and the EM Algorithm, *Proceedings of the 10-th British Machine Vision Conference*, The University of Nottingham, 13-15 September, pp 55-68, ISBN 1-901725-01-X

Nelson, R. C. and H. Samet (1986) A consistent hierarchical representation for vector data, *Computer Graphics*, **4**, pp. 197-206.

Samet, H. (1990) *The Design and Analysis of Spatial Data Structures*, Addison Wesley Publishing Co. Inc., Reading Massachussets, ISBN 0-201-50255