3D datasets segmentation based on local attribute variation

Carla Silva Rocha Aguiar, André Crosnier, Sébastien Druon

3D datasets segmentation based on local attribute variation

C. S. R. Aguiar, S. Druon, A. Crosnier
LIRMM, UMR 5506 Université Montpellier II - CNRS
161 rue Ada, 34000 Montpellier - France
aguiar,druon,crosnier@lirmm.fr

Abstract— **We present a Graph-based method for low-level segmentation of unfiltered 3D data.**

The core of this approach is based on the construction of a local neighborhood structure and its recursive subdivision. The Minimum Spanning Tree (MST) is the graph support used to measure the attribute variation through the region. The subdivision criterion relies on the evidence for a boundary between two partitions, which is detected through MST edge analysis. Although our algorithm converges to a local minimum, our experiments show that it produces segments that satisfy global properties.

We assume that the 3D image is composed of regions homogeneous according to some criterion (color, curvature, etc.), but no assumption about noise, nor spatial repartition/shape of the regions or points is made. Robustness is achieved by choosing the appropriate neighborhood and the analysis of noise impact on the MST construction. We demonstrate the performance of our algorithm with experimental results on real images.

I. PROBLEM STATEMENT

Segmentation is the problem of partitioning the data into meaningful regions. It is usually an intermediate phase, in which objective is mostly a substantial reduction in data volume and to use segmented regions in higher-level processing, such as image recognition, reconstruction, and modelling. When dealing with three-dimensional (3D) images, segments correspond to surfaces or volumes. Recognizing parts on assembly lines, reconstructing a CAD model from an unstructured input data, recognizing physical anomalies from medical 3D images and 3D scene modelling are some applications where segmentation plays a fundamental role.

The main difficulty in developing and evaluating segmentation algorithms is to establish a definition for meaningful region. Different definitions yield to distinct but at the same time correct results [11][4]. In our segmentation algorithm, points belonging to a region have high local similarity and points belonging to different, but neighbor regions, have high local dissimilarity, according to some criterion. 3D image segmentation using local similarity has been considered before [7][1]. The main difference between the present algorithm and its predecessors is that our segmentation algorithm can treat 3D images from different nature, obtained with sensors based on Laser, fringe projectors, CT-scan, MR-scan, SEMs, among others.

In this paper, we are interested in segmenting 3D images taken from the real world. These images are composed of free-form objects, from unknown statistic population (usually non-Gaussian [2]) and significative regions in multiple scales, for example, an image composed by a set of objects with different sizes. These images can be represented by an unorganized sampled point cloud with variable points density (scattered). No information about their structure, nor the topology of the objects presented are supplied by the acquisition system or by the application.

The method proposed relies on the recursive cut of a neighborhood graph. The connexity of this graph embodies the spatial relationships between points, while the weight of each edge represents the local variation of a user defined cutting criterion. From this graph is extracted a Minimum Spanning Tree (MST), which is the support of the segmentation. The use of a MST guaranties the method to converge towards well-conditioned regions in terms of connexity and variations of the segmentation criterion.

Two major contributions of our work are the extension of graph-based segmentation to 3D input data and the fact that our algorithm is not dedicated to one specific cutting criterion. Any locally defined attribute, like color, density or curvature can be used.

The paper will be organized as follow: in section II we give a brief overview of graph-based segmentation methods. In section III we describe the proposed algorithm. Our experimental results using real 3D images are presented and discussed in section IV. Finally, some conclusions and the future works are exposed in section V.

II. RELATED WORK - GRAPH-BASED SEGMENTATION METHODS

Most of low-level segmentation algorithms applied to 3D data are adaptation of methods designed to segment 2D images. Graph-based algorithms have played a particularly important role because they make no assumption about the nature of the objects present in the image and produce well-shaped regions with global properties.

The general principle of graph-based approaches is the construction of a weighted graph $G = (V,E)$, also called *similarity* graph, and then its partition in order to minimize some cost function. These algorithms group points according to their similarity in some attribute space. Points inside a region have a high similarity, while a high dissimilarity is observed between points belonging to neighbor regions. The strategy used to construct, to assign the similarity measure between connected points and to partition the graph G characterize a graph-based algorithm.

Segmentation algorithms based on graph-cuts are the most used in 2D image segmentation, in special normalized cuts (Ncuts), introduced in [10]. These methods minimize the maximum energy (cut) between any two possible partitions of a weighted graph. The minimization is formulated as a generalized eigenvalue problem, and the global minimum of the energy function is guaranteed. The use of graph-cuts in 3D image segmentation is still limited, due mainly to their high computational complexity. Several authors [6] modify graph-cut algorithms in order to segment medical 3D images, where volumetric images can be treated as a set of correlated 2D images or coupled surfaces.

Our work is closely related to segmentation algorithms using Minimum Spanning Trees, introduced in [13]. In [3] and [12] the original method is adapted to segment real 2D image. Noise influence is studied in [12] and algorithm properties and methods comparison is provided in [3]. Edge-based segmentation is exploited in [9] to extract close contours from range images. In [9], the MST is the support to closure refinement, where edge points are given as input of the algorithm.

Compared to other segmentation algorithms based on graph-cuts, our method presents some advantages. First, it runs in time and memory cost nearly in the number of graph edges. Second, the segments have acceptable global properties, even if it converges to the local minimum. Consequently, it can be applied to a larger range of application and to large datasets.

While the approaches using MST [3] and [12] were conceived to segment 2D image, our approach uses a neighborhood graph to make it applicable to segment images from different natures (2D, range, volumetric images, video sequences). And contrary to [9], our work is region-based and the MST is the support to the actual segmentation process.

III. A DESCRIPTION OF THE ALGORITHM

A. Terminology and algorithm overview

We consider a 3D image defined as a set of sampled points $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$. Each element of this set $\mathbf{x}_i = (\mathbf{p}_i, a_i)^T$ is composed of the point coordinates $\mathbf{p}_i \in \Re^3$ and its attribute value $a_i \in \Re$. Brightness, Hue, texture or curvature are examples of such attributes.

From this set of sampled points, we aim at building subsets characterized by the continuity of the attribute values over the subset. The overview of our segmentation algorithm is showed in fig. 1, and it proceeds as follows:

1) *Neighborhood Graph Building*: The spatial neighborhood information of each point is gathered in an undirected graph structure. This graph is referred as the Neighborhood Graph, G_N. The nodes of this graph are the points of the dataset.
2) *Edge Weighting*: To each edge of the neighborhood graph is assigned a weight which is the distance according to the attribute values.
3) *MST Extraction*: From the weighted neighborhood graph is extracted a Minimum Spanning Tree. The edges of this spanning tree link points which are neighbors in both euclidian and attribute spaces.
4) *Recursive cutting*: The final step of our method is the segmentation itself. A hierarchical cutting algorithm recursively splits the regions until they reach homogeneity in attribute space.

B. Neighborhood Graph Building

Neighborhood around a point is a local surface descriptor. In our algorithm, neighborhood is used to obtain points connectivity, and to guarantee that segmented regions are spatially related.

Points density is an unknown variable which must be identified in our approach. This is done by picking randomly some sample

points, and for each sample find its closest neighbor and compute their distance. The density is estimated as being the average distance between closest points. Finally, the sphere radius is defined as $r = N * \sigma$, where N is an input parameter.

The neighboring information is then used to construct an undirected graph G_N, called *Neighborhood Graph* (eq. 1). The sampled points $X = \{\mathbf{x}_i\}$ correspond to the set of vertices or nodes of the graph, $E = \{(\mathbf{x}_i, \mathbf{x}_j)\}$, with $E \subseteq X \times X$, is the set of edges, linking node pairs and they represent the neighborhood relationship.

$$G_N = (\{\mathbf{x}\}, \{(\mathbf{x}_i, \mathbf{x}_j)/\|\mathbf{p}_i - \mathbf{p}_j\| < r\}) \qquad (1)$$

C. Edge Weighting

We apply an edge weighting to the neighborhood graph in order to represent points relations in the chosen attribute space.

The edge weighting consists in assign to each connected pair points, represented by the edge $(\mathbf{x}_i, \mathbf{x}_j)$ in the Neighborhood Graph G_N, the cost $w_{i,j} = \|a_i - a_j\|$, which is the points distance in the attribute space. We obtain then not only connectivity information from the edges of the neighborhood graph, but also attribute variation through surface/volume, explicit by the edges weight.

D. MST Extraction

The next step of our algorithm is to extract a Minimum Spanning Tree in attribute space from the weighted neighborhood graph.

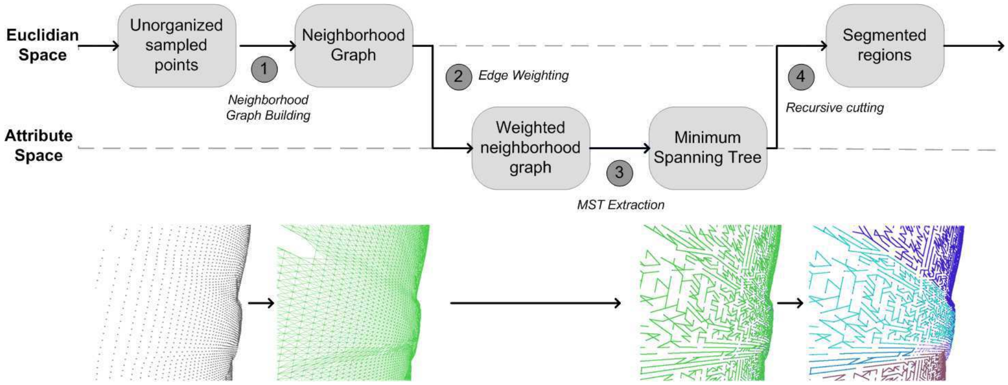


Fig. 1. Segmentation algorithm overview.

The Minimum Spanning Tree is a subgraph of a graph. The MST connects all points (nodes) from the graph forming a tree. All edges presented in the spanning tree are extracted from the original graph.

In a MST, there are no cycles between any two points. This defines a tree, and among all possible spanning trees, the MST is the one where the sum of all costs is minimal.

In our segmentation algorithm, a MST is extracted from the Neighborhood Graph using the classic Prim's algorithm [8]. From the MST algorithm construction, it is always preferred paths in gradient direction and, if any edge $(\mathbf{x}_i, \mathbf{x}_j)$ is removed from the MST, two new independent MSTs are formed. The first property ensures that paths passing through discontinuity regions in attribute space are avoided and noise points and its neighbors are leaf nodes in the MST. From the second property, if nodes of the MST represent points of a given region, removing any edge of the MST create two independent MSTs and, consequently, two subsets of the original set.

Noise is amplified when weighting the edges of the Neighborhood Graph in attribute space, due to the differential nature of the edge cost. Because of that, one can expect a large cost between noise regions and the rest of the samples. In MST algorithm they're naturally placed in leaf nodes. This behaviour is explored in segmentation procedure to filter region segmentation.

E. Recursive cutting

The final step of our algorithm recursively breaks the regions until they reach homogeneity according to the chosen attribute. We used the hierarchical clustering, an unsupervised hierarchical segmentation technique applied specially to segment sampled data from an unknown distribution.

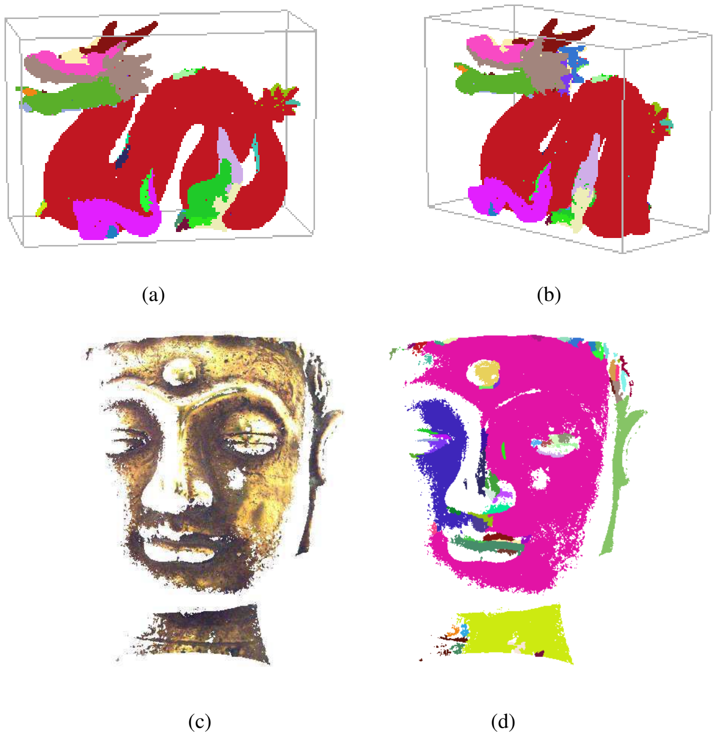
At the initialization, all points belong to the same region and the MST explicits the relation between points in attribute space. The principle of the segmentation procedure is to recursively remove edges from the MST. Once an edge is removed, two new disjoints MSTs are created, and the original region is split in two new regions. The attribute variation through each new region is smaller than the variation through the original one. This process is terminated when homogeneity in attribute space is achieved.

The segmentation algorithm recursively proceeds as follow: it traverses a given MST (representing a region) and it takes the link with biggest cost, $(\mathbf{x}_i, \mathbf{x}_j)^{max}$. Then, a noise test is performed on the link nodes by seeing if one of these points $\mathbf{x}_i$ or $\mathbf{x}_j$ are leaf nodes of the MST. If one of the nodes is a leaf node, smoothing is applied to the link cost and the segmentation procedure is restarted. Otherwise, the similarity function is applied to determine if the region must be split in two or not. In our algorithm, the similarity function is simply a comparison between the biggest cost, $w_{i,j}^{max}$, and a cutting criterion. The cutting criterion is a threshold value and the region is split in two if the biggest cost is above the threshold. Region partition is done by removing the edge $(\mathbf{x}_i, \mathbf{x}_j)^{max}$ of the original MST. Segmentation procedure is repeated until all edges linking the points of a given region (MST) have their cost bellow the threshold value.

The drawback of this technique is that the cutting decision depends upon only one link between two points. It makes the original algorithm sensitive to noise and to data distribution, and it could cause surface/volume over-segmentation. Robustness is achieved in our algorithm by applying the cutting procedure to the MST extracted from the Neighborhood graph.

IV. ALGORITHM EVALUATION

In this section we present the algorithm setup, the nature and characteristics of the images treated and some results.



(a)

(b)

(c)

(d)

Fig. 2. Real range image segmentation.

A. Algorithm setup

Three input parameters are taken in our algorithm: the radius r used to construct the Neighborhood graph, the attribute space one wants to segment the image and the threshold used as cutting criterion.

Attribute space is chosen depending on the information of the input data one wants to segment. Different attributes like color, texture, confidence, curvature, intensity can be used. Color, texture, and confidence descriptors are directly supplied by the acquisition system. When segmenting range images, differential descriptors which provide curvature information such as sphere fitting, the angle between normal vectors and Gauss curvature must be computed. In our algorithm, the normal vectors, fundamental component to compute these geometric descriptors, are estimated by plane fitting using the technique described in [5].

The cutting criterion is a simple threshold that depends on the attribute space and the level of detail one wants to extract from the image. For small threshold values, large surfaces tend to be over-segmented and smaller surfaces are correctly extracted. The opposite is observed for big values. Tuning is necessary to choose the best threshold value but it depends only on the attribute space, and it doesn't depend on the image nature.

B. Results and Discussion

To validate our algorithm and its applicability in a broad range of image segmentation tasks, we took performance measurement over different datasets as video sequence, range and volumetric images.

1) Range image segmentation: The goal of range image segmentation is the partition the input data into surfaces. Input data consists in point coordinates $\mathbf{p}_i = (x,y,z)^T$ and, depending on the acquisition system, color or texture information.

Range images are generally contaminated by heavy noise, different resolution and missing data. Besides that, we don't use any pre-processing algorithm to smooth the surface or to eliminate noise. When curvature descriptors are chosen, the desirable neighborhood radius size, defined by the input parameter N, depends on a trade-off between the conflicting goals of maximum the signal of noise and locating the boundaries accurately.

Figure 2 illustrates some results of range image segmentation. It was used range images taken from different

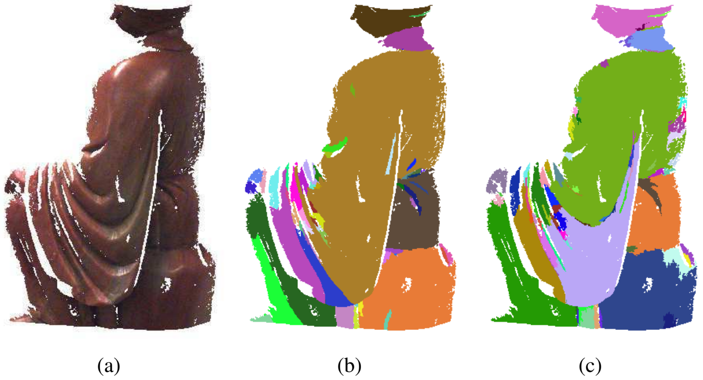


Fig. 3. Influence of the attribute space choice on the segmentation. Original range image (a). Segmented image in (b) angle between normal vector space, and (c) sphere fitting space.

acquisition systems. Figs. 2(a) and 2(b) show the segmented dragon from the Stanford 3D Scanning Repository[1], and figs. 2(d) and 2(d) show the original and segmented range image from our database. Angle between normal vectors was the chosen attribute in segmentation for all images, due to the richness of geometric information of the objects and the same input parameters were used. We use $r = 3 * \sigma$ and cutting threshold of ten degrees.

The segments observed in figure 2 show that our algorithm produces regions with global properties, even if the partition criterion is based on local information. In fig. 2(d) it's observed some small and meaningless surfaces, which are generated especially because of boarder points and missing data. Boarder points have generally smaller density compared to the rest of the image points, and consequently, attribute estimation is less accurate over the entire region, and it is not recognized by our algorithm as noise. A simple region merging algorithm could overcome this problem, but it was not implemented in the present approach.

Figure 3 outlines that the image can be segmented according different attribute spaces and the influence of the attribute choice is studied. For that, we take the same image and segment it according to different attribute spaces: the angle between the normal vectors and the sphere fitting. The original and segmented images are showed in fig. 3. Both segmentation procedures took the same radius $r = 2 * \sigma$ to construct the Neighborhood graph. Although both attributes provide local information about surface curvature, we can observe that our algorithm segmented the image into different regions. This is due to the attribute's nature and the influence of noise on attribute estimation. The main problem of these attributes is that they're computed around a small region and when derivatives are computed, noise present in the input and in normal estimation are amplified. We can see that sphere fitting estimation is more affected by noise, since segmentation in this attribute space created many small and meaningless regions.

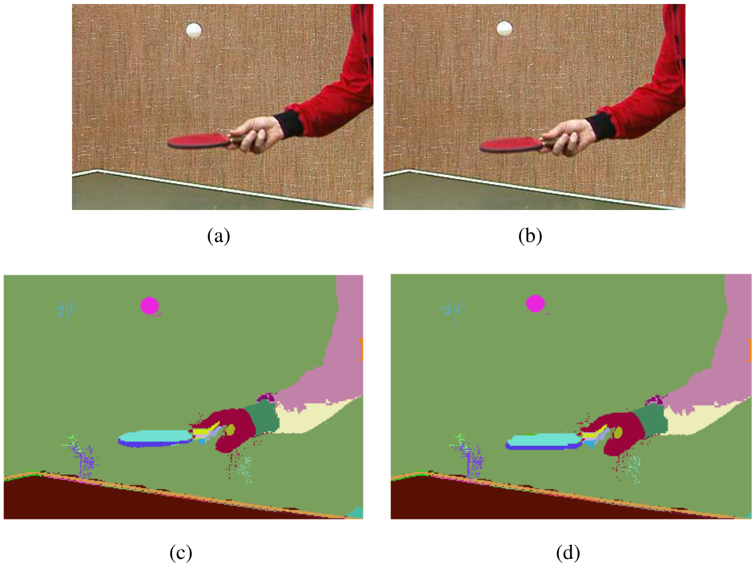
2) Volumetric image and video sequence segmentation: Volumetric images are commonly obtained from devices such as CTs and MRI. It is composed by a set of 2D images, grouped in layers in order to form the volume. Point coordinate $\mathbf{p}_i = (x, y, z)^T$ is directly obtained by image pixel coordinate and the relation between successive image layers. Gray level embeds material properties. Segmentation breaks the input image into volumes, where each region has an homogeneous material property.

If we consider time as a third dimension in our neighborhood graph, the method can be extended to video sequence segmentation. The initial dataset is indeed composed of all the pixels from all the video frames labelled with a timestamp. The problem with this approach is the sampling frequency inconsistency between spatial and temporal dimensions. To overcome this difficulty, we ensure that the time step is set to 1. This extension can only be envisaged when dealing with offline video segmentation because all the pixels must be available for the neighborhood graph construction.

Fig. 4 show the original and segmented frames of a video sequence [2]. Note that there is considerable variation in the background color. This kind variability is handled in our algorithm by choosing a large neighborhood size. We used $r = 4 * \sigma$. The drawback of choosing a large neighborhood is that, in this case, our algorithm doesn't guarantee to obtain compact regions, and this effect is observed in figs. 4(c) and

[1] From http://graphics.stanford.edu/data/3Dscanrep/, Stanford 3D Scanning Repository.

[2] From http://sampl.ece.ohio-state.edu/data/motion/tennis, Ohio State University.


(a)

(b)

(c)

(d)

Fig. 4. Video sequence segmentation. Original video frames(a) and (b). Segmented images (c) and (d).

4(d). This behaviour is not critical, once the resulting regions are generally well bounded and have global properties and, consequently, our segmentation algorithm can be used to track objects in a video sequence.

V. CONCLUSIONS AND FUTURE WORKS

We have presented an algorithm for segmenting 3D images into homogeneous regions, according to an attribute. The important properties of our algorithm are that it can segment the image in different attribute spaces, it can extract different level of detail in the image, and the fact that no a priory knowledge is made about the object topology, or structure, or points distribution. One advantage of this segmentation method when dealing with range image is its ability to deal with noisy or incomplete datasets. This makes it particularly useful when dealing with raw data from a 3D scanning system.

Further work is ongoing towards video stream segmentation for real time object tracking. On-the-fly segmentation supposes a new definition of the neighborhood relationship and imposes time constraints. Concerning range image segmentation, we would like to use this approach to solve 3D surface fitting problems.

REFERENCES

[1] P. J. Besl and R. C. Jain. Segmentation through variable-order surface fitting. *IEEE Trans. Pattern Anal. Mach. Intell.*, 10(2):167–192, 1988.

[2] K. L. Boyer, M. J. Mirza, and G. Ganguly. The robust sequential estimator: A general approach and its application to surface organization in range data. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16(10):987–1001, 1994.

[3] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *Int. J. Comput. Vision*, 59(2):167–181, 2004.

[4] A. Hoover, G. Jean-Baptiste, X. Jiang, P. J. Flynn, H. Bunke, D. B. Goldgof, K. Bowyer, D. W. Eggert, A. Fitzgibbon, and R. B. Fisher. An experimental comparison of range image segmentation algorithms. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(7):673–689, 1996.

[5] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Surface reconstruction from unorganized points. *Computer Graphics*, 26(2):71–78, 1992.

[6] K. Li, X. Wu, D. Z. Chen, and M. Sonka. Optimal surface segmentation in volumetric images-a graph-theoretic approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(1):119–134, 2006.

[7] G. Osorio, P. Boulanger, and F. Prieto. An experimental comparison of a hierarchical range image segmentation algorithm. In *CRV '05: Proceedings of the The 2nd Canadian Conference on Computer and Robot Vision (CRV'05)*, pages 571–578, 2005.

[8] R. Prim. Shortest connection networks and some generalizations. *Bell System Technical Journal*, 36:1389–1401, 1957.

[9] A. D. Sappa. Unsupervised contour closure algorithm for range image edge-based segmentation. *IEEE Transactions on Image Processing*, 15(2):377–384, 2006.

[10] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[11] R. Unnikrishnan, C. Pantofaru, and M. Hebert. Toward objective evaluation of image segmentation algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):929–944, June 2007.

[12] Y. Xu, V. Olman, and E. C. Uberbacher. A segmentation algorithm for noisy images. In *IJSIS '96: Proceedings of the 1996 IEEE International Joint Symposia on Intelligence and Systems*, page 220, Washington, DC, USA, 1996. IEEE Computer Society.

[13] C. T. Zahn. Graph theoretical methods for detecting and describing gestalt clusters. *IEEE Trans. Comput.*, 20:68, 1971.