 Cytometry 35:11–22 (1999)

Segmentation of Tissue Architecture by Distance Graph Matching

Jan-Mark Geusebroek,[1,2]* Arnold W.M. Smeulders,[1] Frans Cornelissen,[2] and Hugo Geerts[2]

[1]Department of Computer Science, Faculty of Science, University of Amsterdam, Amsterdam, The Netherlands
[2]Life Sciences, Janssen Research Foundation, Beerse, Belgium

Received 17 June 1998; Revision Received 24 August 1998; Accepted 31 August 1998

Background: Characterization of tissues can be based on the topographical relationship between the cells. Such characterization should be insensitive to distortions intrinsic to the acquisition of biological preparation. In this paper, a method for the robust segmentation of tissues based on the spatial distribution of cells is proposed.
Materials and Methods: The neighborhood of each cell in the tissue is modeled by the distances to the surrounding cells. Comparison with an example or prototype neighborhood reveals topographical similarity between tissue and prototype. Processing of all cells in the tissue extracts the regions with tissue architecture similar to the given example.
Results: Comparison with other topographical-segmentation methods shows that the proposed method is better suited for partitioning tissue architecture. As an example, the quantification of the structural integrity in rat hippocampi after ischemia is demonstrated. In contrast to other methods, the algorithm correlates well with expert evaluation.
Conclusions: The present method reduces the nonbiological variation in the analysis of tissue sections and thus improves confidence in the result. The method can be applied to any field where regular patterns have to be detected, as long as the directional distribution of neighbors may be neglected. Cytometry 35:11–22, 1999. © **1999 Wiley-Liss, Inc.**

Key terms: cell topography; cellular sociology; nearest-neighbor graphs; tissue; segmentation; rat hippocampus

Quantitative morphological analysis of fixed tissue plays an increasingly important role in the study of biological and pathological processes. Specific detection issues can be approached by classical staining methods, enzyme histochemical analysis, or immunohistochemical processes. The tissue can not only be characterized by the properties of individual cells, such as staining intensity or expression of specific proteins, but also by the geometrical arrangement of the cells (1–3). Interesting tissue parameters are derived from the topographical relationship between cells. For instance, topographical analysis in tumor grading can significantly improve routine diagnosis (4–6). Studies of growing cancer cell lines have revealed a non-random distribution of cells (7,8). Partitioning of epithelial tissue by cell topography is used for quantitative evaluations (9). We propose a new method for the partitioning of tissues. As an example, structural integrity of hippocampal tissue after ischemia will be examined.

As a first step, tissue parts of interest have to be segmented into cell clusters. Segmentation of cell clusters can be based on distances between the center of gravity of the cells. The recognition of tissue architecture is then reduced to determining borders of point patterns. The problem traced as such can be solved by the application of neighbor graphs, and partitioning them.

The Voronoï graph is often applied as a modeling tool for point patterns (4,6,7,10). The definition of the Voronoï graph is given by polygons Z(p), where each polygon defines the area for which all points are closer to marker p than to any other marker (11). A polygon Z(p) is called the zone of (geometrical) influence of p. Neighboring markers to p are defined by the set of all markers for which the zone of influence touches that of p. Such a tesselation of the plane depends on the spatial distribution of the cell markers. Cluster membership is determined by evaluation of geometrical feature measurements on the zones of influence (12).

Rodenacker et al. (9) used the Voronoï graph for partitioning epithelial tissue. Segmentation was obtained by propagating the neighbors from the basal layer of the epithelial tissue to the surface. Borders between basal, intermediate, and superficial areas were determined by examining the occupied surface of propagation. In this way, every third of the total area of the Voronoï graph was assigned to one of the regions, yielding three regions with approximately similar areas in terms of zones of influence.

*Correspondence to: Jan-Mark Geusebroek, Faculty of Science, Department of Computer Science, University of Amsterdam, Kruislaan 403, 1098 SJ Amsterdam, The Netherlands.
E-mail: mark@wins.uva.nl

As discussed elsewhere (7,13), the Voronoï graph is sensitive to detection errors. Removal or insertion of one object will change the characteristics of the Voronoï graph. A second drawback is that Voronoï's graph is ill-defined at cluster borders. This makes the Voronoï graph unsuited for robust segmentation of tissue architecture.

Another option for the recognition of point patterns is a modification of the Voronoï graph: the k-nearest neighbor graph (14–16). The neighbors of a point p are ordered as the nearest, second-nearest, and up to k^{th}-nearest neighbor of p. The k-nearest neighbor graph is defined by connecting each point to its k-nearest neighboring points (11). The strength of each connection is weighted by the distance between points. Similarity between k-nearest neighbor graphs is determined by comparing the graphs extracted from detected point patterns with prototype k-nearest neighbor graphs.

In Schwarz and Exner (16), the distance distribution to one of the nearest neighbors was used for the separation of clusters from a background of randomly disposed points. The main drawback is that not all patterns can be discriminated by considering only one specific k-nearest neighbor distance.

Lavine et al. (14) used sequences of sorted interpoint distances extracted from noisy point images to match the image with one of a set of prototype patterns. Similarity between prototype and point set is based on a rankwise comparison. From the two sorted interpoint distance vectors, the corresponding (relative-) difference vector is calculated. The number of components which exceeds a given threshold is used for discrimination between patterns. A major disadvantage of the rankwise comparison is that all components have to be detected. When the nearest neighbor is missed in the detection, the first one in rank is compared with the second one. Thus, failure to detect one cell results in poor similarity.

Automatic segmentation of tissue architecture is difficult because biological variability and tissue preparation have a major influence on the tissue at hand. The detection and classification of individual cells in the tissue is prone to error. Although most authors (4,7,13) were aware of the lack of robustness in the quantification of tissue architecture, little effort was made to incorporate uncertainty of cell detection in tissue architecture quantification methods. Lavine et al. (14) showed that the k-nearest neighbor graph is well-suited for point pattern recognition under spatial distortions, but the method used is not able to anticipate cell detection errors.

In this paper we present a robust method for tissue architecture segmentation, based on the k-nearest neighbor graph. A sequence comparison algorithm is used to allow missing or extra detected cells in the detected point set. Uncertainty in cell classification is incorporated into the matching process. Experiments show that the robustness of the method presented is superior to that of existing methods.

The method is demonstrated by segmentation of the CA region in rat hippocampi, where structural integrity of the CA1 cell layer is affected by ischemia. The correlation between manual scoring and automatic analysis of CA1 preservation is shown to be excellent.

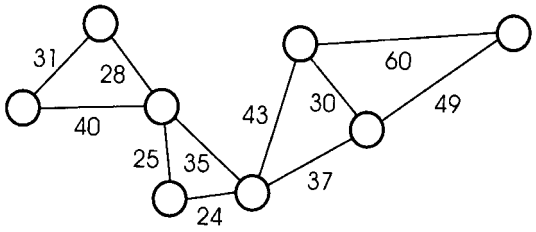

FIG. 1. Example of a k-nearest neighbor graph. The nodes represents cells in tissue, while the edges represent their relation. The relations in this graph are given by the two nearest neighboring cells, and edges are weighted by the distance between the cells.

MATERIALS AND METHODS

Hippocampal Tissue Preparation

Rat brains were fixed by intracardiac perfusion with diluted Karnovsky's fixative (2% formaldehyde, 2.5% glutaraldehyde in Sörensen's phosphate buffer; pH 7.4). They were immersed overnight in the same fixative. Coronal vibratome sections of the dorsal hippocampus were prepared stereotaxically 3.6 mm caudally to the bregma (Vibratome 1000, TPI, St. Louis, MO). Slices (200 μm) were postfixed with 2% osmium-tetroxide, dehydrated in a graded ethanol series, and routinely embedded in Epon. Epon sections were cut at 2 μm and stained with toluidine blue.

Image Acquisition and Software

Images were captured by a CCD camera (MX5, Adimec, Eindhoven, The Netherlands), which is a 780 × 576 video frame transfer CCD with pixel size 8.2 × 16.07 μm^2, operating at room temperature with auto gain turned off. The camera was mounted on top of an Axioskop in bright-field illumination mode (Carl Zeiss, Oberkochen, Germany). The microscope was equipped with a scanning stage for automatic position control (stage and MC2000 controller, Märzhäuser, Wetzlar, Germany). The scanning stage was calibrated for a 10× magnification, and adjacent 512 × 512 images were captured to ensure that complete hippocampi were scanned. Typical composite image sizes were 6,144 × 4,096 pixels, or 4.94 × 3.30 mm^2. For image processing, the software package SCIL-Image version 1.4 (TNO-TPD, Delft, The Netherlands) was used on an O2 workstation (SGI, Mountain View, CA). The package was extended with the distance graph matching algorithm.

K-Nearest Neighbor Graph

Consider an image of a tissue containing cells. Detection of cells in the image will result in m markers at possible cell locations. Let V be the set of m detected cell markers, $V = [v_1, v_2, \ldots v_m]$. The elements in V are called vertices or nodes. A graph G(V, E) (Fig. 1) defines how elements of V are related to one another. The relation between the vertices is defined by the set of edges E, in which the

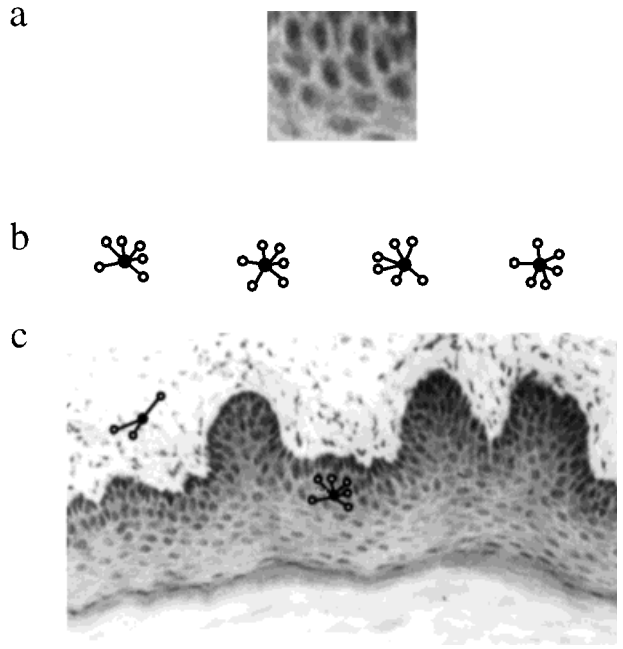

FIG. 2. Extraction of tissue architecture. A typical relationship around a cell is obtained from an example of the tissue of interest (**a**). The prototype k-nearest neighbor graph is derived from distances to cells (**b**). All prototypes shown are considered equal to fit deformed tissue parts. Further freedom is given by a certain elasticity of the edges in the prototype graph. Extraction of the tissular architecture proceeds by fitting the prototype graph on each cell and its neighborhood in the tissue (**c**). Within the similar tissue parts, the graph will fit. Outside these regions, matching is limited to only one or two edges. In order to safeguard against cell detection errors, not all edges in the prototype have to fit the cellular neighborhood.

elements e_{ij} connect the vertices v_i to v_j. A weighted graph is defined by the graph G(V, E), where a value is assigned to each edge e_{ij}.

The k-nearest neighbor graph of a node v is defined as the subset of k vertices closest to v. The edges between v and the neighboring vertices are weighted by the Euclidian distance, or $N_v^k = \{d_1, d_2, \ldots, d_k | \; d_i = \text{dist}(v, v_i), d_i < d_{i+1}\}$. Taking $k = 1$ for all $v \in V$ results in the nearest neighbor graph, in which each cell is connected to its closest neighbor.

The average edge length in the k-nearest neighbor graph gives a measure of scale of the pattern of cells. Division of all distances d_i in a k-nearest neighbor graph by the average of all distances in the graph, $\bar{d}$, normalizes the graph for scale, i.e., $\tilde{d}_i = d_i/\bar{d}$.

Distance Graph Matching

Point patterns of interest were extracted from the k-nearest neighbor graph. As an example, consider Figure 2. A regular structured tissue was assumed, consisting of cells regularly distributed over the tissue. Such a point pattern reveals an equally spaced layout everywhere within the tissue borders. The surrounding of each cell belonging to the pattern can be modeled by the neighborhood of one single cell (Fig. 2). The k-nearest neighbor graph of a typical pattern cell gives a characterization of the point pattern of interest. After selection of a typical cell, the pattern is given by a prototype k-nearest neighbor graph, with distance set $P = \{p_1, p_2, \ldots, p_k\}$, where p_i denotes the prototype distances. Acceptance or rejection of a detected object as belonging to the cell cluster of interest is based on comparison of the observed k-nearest neighbor distances N_v^k, to the prototype defined by the characteristic distances to the neighbors in P.

Distance Graph Comparison

The difference between observation and prototype set is expressed by the replacements necessary to match the prototype with the observation. This is referred to as dissimilarity between sets (17). For example, consider for simplicity the discrete observed set {3, 10, 11, 15, 20, 20, 21, 25} and prototype {5, 5, 10, 10, 20, 20}. When disregarding the last distances in the observation (21 and 25), two substitutions ($3 \rightarrow 5$, $11 \rightarrow 10$), one insertion (5), and one deletion (15) transform the observed distance set into the prototype. So there are four modifications between prototype and observation. The extra distances at the end of the observed set are necessary for expanding the comparison when elements are deleted in the beginning of the set. Without these extra elements, deletion of one item at the beginning of the set implies the addition of an item at the end of the set. There will be no need for addition when there is a cell at the correct distance. Therefore, the amount of elements in the observation *l* should be larger than the prototype length k to allow for expansion in the comparison.

A cost is assigned to each type of replacement. Let c_i be the cost for insertion, c_d the cost for deletion, c_s the cost for substitution, and c_m the cost for matching. In the example, 11 is closer to 10 than 3 is to 5, which can be reflected in their respective matching costs. The minimum total cost t, necessary to transform the observed set into the prototype, gives the similarity between the sets. The minimum cost is obtained by using a string matching algorithm (17) (see Appendix).

The lowest possible value for the cost t is obtained when both sets are equal. The amount of replacements is zero, and thus the cost is zero. An upper bound for the cost necessary to match two sets is obtained when all elements are replaced. In this case, either all elements are inserted at the beginning of the set, or all elements are substituted, depending on the respective costs. The upper bound is then given by $t_{upper} = k \min(c_i, c_s)$. Normalization of the minimum total cost gives a correspondence measure, indicating how well the observed pattern matches the prototype, i.e.,

$$C = \frac{t_{upper} - t}{t_{upper}} \times 100\%. \qquad (1)$$

Discrimination between two known point patterns, cluster and background, can be based on example and counterexample. Consider the observed k-nearest neighbor graph N_v^k, the prototype P describing the pattern of

interest, and a prototype B characterizing the background pattern. When elements in background B match elements in P, the cost t_{backgr} related to matching P with B is less than the upper bound for the minimum cost. Then, discrimination between the two patterns is enhanced by normalizing the correspondence to the cost given by matching P with background B, or

$$C' = \frac{t_{backgr} - t}{t_{backgr}} \times 100\%. \tag{2}$$

Note that C′ can be negative for patterns which neither correspond to the foreground prototype nor to the background prototype. The extension to multiclass problems can be made by considering prototype P for the class of interest, and prototypes $B_1, B_2, \ldots, B_n$ for the remaining classes. Matching P with each of the prototypes B_i gives the correspondences between the pattern of interest and the other patterns. The pattern B_i which is most similar to P results in the lowest matching cost, which should be used for normalization.

Cost Functions

The total cost depends on the comparison between each of the individual elements of N_v^k and P, and thus the replacements necessary to match them. The replacement operations are given by insertion (cost c_i), deletion (cost c_d), substitution (c_s), and match (c_m).

The cost for matching c_m is zero when the two distances are equal. The difference between two distances is defined as their relative deviation, or $\delta = |d_i - p_j|/p_j$. Here, d_i denotes the observed distance and p_j the prototype distance with which to compare. Robustness against spatial distortion is obtained by allowing a percentage deviation α in the comparison of distances (14). In this case, two distances are considered equal as long as their relative deviation is smaller than the tolerance α. A minimum value for α is given by the distance measurement error.

When the deviation percentage between two distances is higher than α, their correspondence is included in the matching cost. The correspondence C then depends on the total distance deviation between the compared elements. The matching cost is taken linearly proportional to the distance deviation, or

$$c_m = \begin{cases} 0 & \text{if } \delta \le \alpha \\ (\delta - \alpha)\dfrac{c_s}{c_s - \alpha} & \text{if } \alpha < \delta < c_s \\ c_s & \text{otherwise.} \end{cases} \tag{3}$$

The cost for matching is c_s if $\delta \ge c_s$, which is equivalent to a substitution operation.

For our case, cell detector properties determine the cost for insertion. For a sensitive detector, the probability to miss a cell is low. As a consequence, the cost for insertion should be high compared to deletion. Alternatively, a low-sensitive cell detector will overlook cells, but fewer artifacts will be detected. Thus, the costs for insertion should be low relative to deletion. The insertion cost is therefore tuned to the cell detector performance, or

$$\frac{c_i}{c_d} \propto \frac{\#A}{\#M}. \tag{4}$$

Here, #A denotes the estimate of the average amount of artifacts detected as cells, and #M denotes the estimate of the average amount of missed cells.

The deletion cost is derived from object features. A probability distribution can be obtained from well-chosen measurements, e.g., the contour ratio, on a test set of objects. Afterwards, the probability $P(v_i)$ for object v_i being a cell is extracted from the measured distribution. When an object has a low probability of being a cell, the object should be deleted. Therefore, rather than considering a fixed deletion cost, the probability of an object being a cell determines the deletion cost for that object, or

$$c_i(v_i) \propto P(v_i). \tag{5}$$

As a result, the correspondence measure for the object under examination is only slightly affected by the deletion of artifacts. The rejection of detected objects as being artifacts can be based on both cell probability $P(v_i)$ and the correspondence C of the object to the cluster prototype.

Evaluation of Robustness on Simulated Point Patterns

Four algorithms, based on the Voronoï graph, nearest neighbor distance, Lavine's method for matching, and the proposed distance graph matching, were tested in simulations. The segmentation performance was measured as a function of the input distortion. The input consisted of a foreground point pattern embedded in a background pattern, distorted by some random process.

For the simulations, two arbitrarily chosen patterns were generated. A hexagonal point pattern was embedded in a random point pattern with the same density, and the same pattern was placed in a hexagonal pattern with half the density (Fig. 3). Artificial distortion was added to the sets by consecutive random removal, addition, and displacement of points. The distortion was regulated from 0% up to a maximum, resulting in a noisy realization of the ideal patterns. By removing points, the algorithm is tested for robustness against missing cells. Addition of points reveals robustness of the algorithm against false cell detections. Robustness against spatial distortion is examined by means of point displacement. Each one of the four methods was tested for robustness against the given distortions. The combination of removal and displacement of points shows robustness against touching cells. The other combinations show the interaction of distortions on robustness.

The segmentation performance indicates how well the foreground pattern was discriminated from the background points. It was measured as function of the distor-

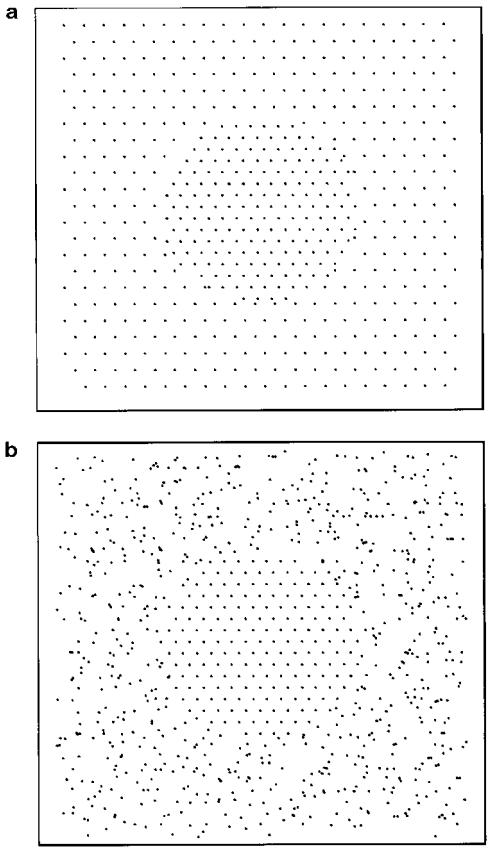

FIG. 3. Point patterns used for the experiments. **a:** A regular pattern inside a regular pattern with half the density. **b:** A regular pattern inside a random pattern with the same density.

tion. The performance of the various algorithms was measured as one minus the ratio of the false negatives combined with the ratio of false positives, or

$$P = 1 - \frac{\#F_b}{\#Truth_f} - \frac{\#B_f}{\#Truth_b}. \quad (6)$$

Here, $\#F_b$ denotes the number of foreground markers classified as background, $\#B_f$ denotes the amount of background markers classified as foreground, and $\#Truth_f$ and $\#Truth_b$ denote the true number of foreground and background markers, respectively, in the distorted data set.

Algorithm Robustness Evaluation

For the experiments, the area of the influence zones in the Voronoï graph was thresholded (18) in order to partition the test point patterns. The thresholds were chosen such that 10% distortion on the distance to the nearest neighbors was allowed for the undistorted foreground pattern. This yields calculation of the minimum and maximum areas for scaled versions of the pattern, with scaling factors 0.9 and 1.1.

With regard to the nearest-neighbor distance, thresholds were taken such that 10% perturbation in the nearest-neighbor distance was allowed, determined in the undistorted foreground pattern.

The method given by Lavine et al. (14) was tested for $k \in \{5, 10, 15, 20, 25\}$. Implementation of this method was achieved by using the distance graph matching algorithm. Examples of both foreground and background pattern were used for discrimination. Costs for insertion and deletion were taken as infinity ($c_i = c_d = \infty$); thus, only substitutions or matches were allowed. The allowed perturbation in the distances was set at 10% ($c_s = \alpha = 0.1$). The correspondence C' (Eq. 2) was thresholded at 50%.

Experiments for the proposed distance graph matching method were taken with prototype length $k \in \{5, 10, 15, 20, 25\}$. In order to allow the string matching to expand, the amount of observed elements considered for matching was twice the length of the prototype set ($l = 2k$). Examples of both foreground and background pattern were used for discrimination. Substitution of cells was not allowed, except as a deletion followed by an insertion operation. This can be achieved by taking the cost for substitution equal to the sum of the costs for insertion and deletion ($c_s = c_i + c_d = c$). The costs for insertion and deletion were taken as equal. The allowed perturbation in the distances was taken to be 10% ($c = \alpha = 0.1$). The correspondence C' (Eq. 2) was thresholded at 50%. This way, parameters were set to permit fair comparison between the four methods for tissue architecture segmentation.

Robustness for Scale Measure

In order to investigate the influence of distortions on the scale normalization measure, the measure was tested in the simulations. The normalization factor $\bar{d}$, the average neighbor distance, was calculated under addition, removal, and displacement of points. The percentage error to the initial scale measure, $\bar{d}$ for 0% distortion, was measured as function of the distortion. The amount of neighbors k considered for calculation of the scale measure was taken to be $\{1, 5, 10, 15\}$.

Cell Detection

Cell domes were extracted from the hippocampal images by grey-level reconstruction (19), resulting in a grey-value image containing the tops of all mountains when considering the input image as a grey-level landscape. From the dome image, saturated transparent parts were removed, and the remaining objects were thresholded. The results contained cell bodies, neurite parts, and artifacts. An opening was applied to remove the neurite parts. After labeling, the center of gravity of each object was calculated and used for determination of the k-nearest neighbor graphs. The reciprocal contour ratio (1/cr) was used as a measure for cell probability (Eq. 5).

Hippocampal CA Region Segmentation

Segmentation of the CA region was obtained by supervised selection of an example region. An arbitrary section,

unaffected by ischemia, was taken and, after cell detection, one of the cells in the CA1 region was manually selected. The neighborhood of the selected cell was used as a prototype for segmentation. No counter (background) example was taken. Each of the four algorithms was used for segmentation of the CA region. Parameters for segmentation were derived from the example neighborhood to permit fair comparison between methods.

Thresholds for the area of the influence zones in the Voronoï graph were derived from the example, such that 35% distortion on the distance to the nearest neighbor was allowed.

For the nearest-neighbor method, thresholds were taken such that 35% distortion on the nearest-neighbor distance in the example was allowed.

The method of Lavine et al. (14) was implemented by using the distance graph matching algorithm. Costs for insertion and deletion were taken as infinity ($c_i = c_d = \infty$), allowing only substitutions or matches with 35% tolerance ($c_s = \alpha = 0.35$). The deletion cost for individual objects was adjusted by the cell probability, derived from the contour ratio. For graph matching, 15 neighbors were taken into account. A cell was considered as a cluster cell when the similarity between distance graph and prototype was at least 50%.

For the distance graph matching method, substitution of cells was not allowed, achieved by setting $c_s = c_i + c_d = c$. The substitution cost was tuned to allow for 35% distortion in the distances, from which the last 25% was included in the correspondence measure ($c = 0.35$, $\alpha = 0.1$). After visual examination of the detector performance, the insertion cost was set at twice the deletion cost. The deletion cost for individual objects was adjusted by the cell probability, derived from the contour ratio. For distance graph matching, 15 neighbors were taken into account. Matching was allowed to expand to twice the amount of neighbors in the prototype ($l = 2k$). A cell was considered as a cluster cell when the similarity between distance graph and prototype was at least 50%.

RESULTS

Algorithm Robustness Evaluation

Figure 4 shows the results of the performance of the algorithms on the simulated point patterns, where 0% performance corresponds to random classification of the markers. The distortion for removal and addition of points is given as the percentage of points removed or added. For displacement of points, the distortion is given as percentage of displacement up to half the nearest neighbor distance (100%) of the undistorted hexagonal foreground pattern. When the distortion in displacement reaches 100%, the hexagonal pattern has become a random pattern, indistinguishable from the random background pattern (Fig. 4f). The optimum performances which can be reached for the three types of distortion are shown in Figure 4b,d,f. In those cases, the segmentation result corresponds to correct classification of all (remaining) markers. The results of the combined experiments are examined for interaction between the different kinds of distortions, and their relation with the individual performances.

The behavior of the algorithms under all distortions remains similar for both test patterns. This suggests that the performance of the different methods is insensitive to the type of test pattern.

For addition and displacement of points, the minimum and maximum performance over the 25 simulation trials remains within 20% from the average. For removal of points, the minimum and maximum performance was within 40% from the average for the Voronoï, Lavine et al. (14), and distance graph matching methods. The nearest-neighbor method shows a deviation of 60% from the average for removal of points, which is due to the normalization of the performance measure to the amount of markers (Eq. 6).

Figure 4a–d reveals that thresholding the area of influence in the Voronoï graph is inadequate in determining cluster membership when cell detection is not reliable. No point can be removed or added without changing the Voronoï partition for all (Voronoï) neighbors surrounding the removed or added point. A second drawback is the high initial error of 20% and 35%, respectively. Under displacement of points (Fig. 4e,f), segmentation based on the Voronoï graph is shown to be robust. Figure 4f reveals the bias (100% distortion, 10% performance) for the Voronoï graph at the image border. Points near the image border are all (correctly) classified as background due to their deviation from the normal area of influence, resulting in a better than random classification for the indistinguishable fore- and background. Experiments for the Voronoï method performed with thresholding the deviation on the nearest-neighbor distance at 5% give only marginally better performances (data not shown). For the combination of displacement and removal, the resulting segmentation error showed both factors to be additive below 15% removal (data not shown). Similarly, for the displacement and addition of points, the combined error was shown to be the addition of errors caused by applying each distortion separately. The performance under removal and addition of points is only slightly better than the addition of the individual errors.

Segmentation based on the nearest-neighbor distance behaves like the optimum when distorted by removal of points (Fig. 4a,b). Under the condition of addition of points (Fig. 4c,d), performance is as bad as with the Voronoï method. Since 10% distortion on the nearest-neighbor distances is allowed, the method performs well up to 10% displacement (Fig. 4e,f). As shown elsewhere (16), segmentation based on one of the other k-nearest neighbors is able to improve the discrimination between patterns. Behavior of the method under distortions for higher k remains similar to the results shown for $k = 1$. The performance for the combinations removal-addition and removal-displacement was completely determined by addition and displacement (data not shown), respectively. As can be expected from Figure 4a,b, the influence of removal of points may be neglected. For the combination of addition and displacement of points, the effect on the

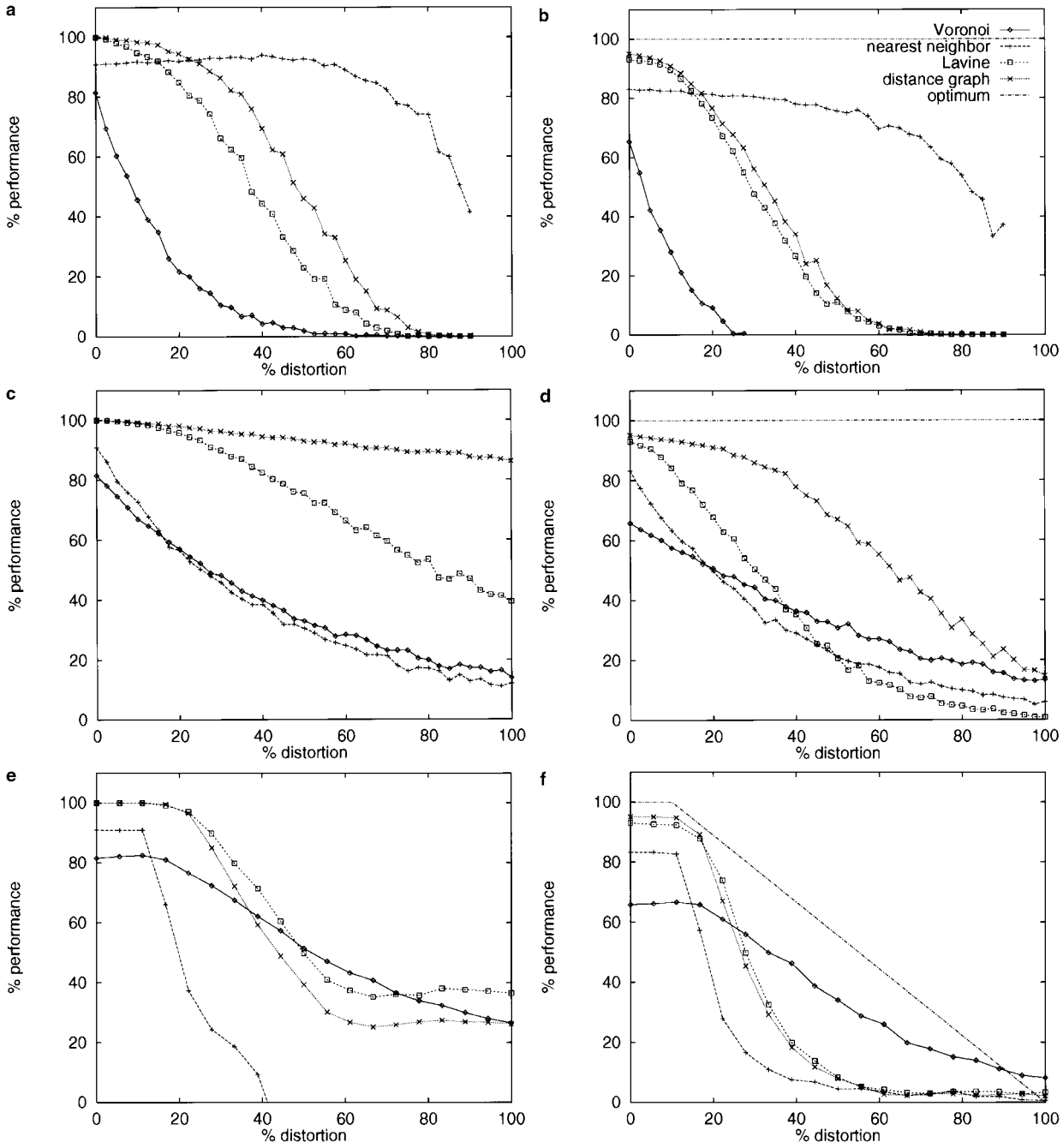

FIG. 4. Average segmentation performance is plotted as function of the distortion. Each point represents the average performance over 25 trials for the given percentage of distortion. For the method of Lavine et al. (14) and the distance graph matching method, results for $k = 10$ are shown. **a:** Point removal, hexagonal background. **b:** Point removal, random background. **c:** Point addition, hexagonal background. **d:** Point addition, random background. **e:** Point displacement, hexagonal background. **f:** Point displacement, random background.

segmentation error is the addition of the errors caused by each distortion separately.

For the method of Lavine et al. (14), the results are shown for $k = 10$. The initial segmentation error between the test point patterns (Fig. 4a,b) is smaller than with both the Voronoï and nearest-neighbor method. Taking more neighbors into account clearly results in better discrimination between point patterns. The performance under removal of points degrades faster than the nearest neighbor segmentation (Fig. 4a,b), while the performance for addition of points (Fig. 4c,d) degrades less severely for small distortions. The tolerance for spatial distortion is improved in comparison to the nearest-neighbor method. Analysis based on higher neighborhood sizes ($k \in \{15, 20, 25\}$) shows that the performance for removal and addition of points degrades faster, whereas the performance im-

proves under the condition of displacement of points. Additionally, the initial error increases with a few percentages. For k = 5, segmentation performance is comparable, except for the initial error, which increases a few percentages. The error due to both the combinations removal-displacement and addition-displacement was shown to be almost perfectly additive (data not shown). For the combination of addition and removal of points, the error due to removal is counteracted by the addition of points for large distortions.

The distance graph matching method performs slightly better than the method of Lavine et al. (14) for removal of points (Fig. 4a,b). Under the condition of point addition, the distance graph matching method is clearly superior. The initial error in the discrimination between both hexagonal foreground and background is zero for both the distance graph method and that of Lavine et al. (14). For the discrimination between hexagonal foreground and random background, the initial performance for the distance graph matching is better than with the method of Lavine et al. (14). Performance for a small neighborhood size is comparable to the performance with the method of Lavine et al. (14) (k = 5). For large neighborhood sizes ($k \geq 15$), performance for removal and addition degrades faster, but remains better than with the method of Lavine et al. (14). Under displacement of points, the performance increases for high k. Additionally, the initial error increases a few percentages. The performance for the combined distortion from addition and displacement of points was shown to be completely determined by the point displacement (data not shown). For removal and addition, the error due to removal was reduced by the random addition of points for severe distortions. The combination of removal and displacement was shown to be better than the addition of the respective errors.

From these experiments, it can be concluded that both thresholding the area of influence in the Voronoï graph and thresholding the distance to one of the nearest neighbors are not suitable for robust segmentation of tissue architecture. The experiments undertaken show the instability of the Voronoï graph for detection errors. The Voronoï graph is certainly useful for determination of neighbors (6), but more robust parameters can be estimated from the Euclidian distance between these neighbors (20). The proposed distance graph matching algorithm indeed has a better performance under detection errors than the method of Lavine et al. (14). Therefore, the distance graph matching method is more suitable for use in the partitioning of tissue architecture.

Robustness for Scale Measure

Robustness of scale normalization was tested on both artificial data sets. Results for k = 10 on the hexagonal-hexagonal data set are shown in Figure 5. The result for k = 1 degrades for addition and displacement of points, while removal of points is more stable. The results for k = 5 and k = 15 are almost identical to the results shown for k = 10. The results for the hexagonal-random data set are almost identical to the hexagonal-hexagonal results for $k \in \{5, 10, 15\}$. The experiment shows that the average k-nearest neighbor distance is useful in normalization for scale when taking k large enough.

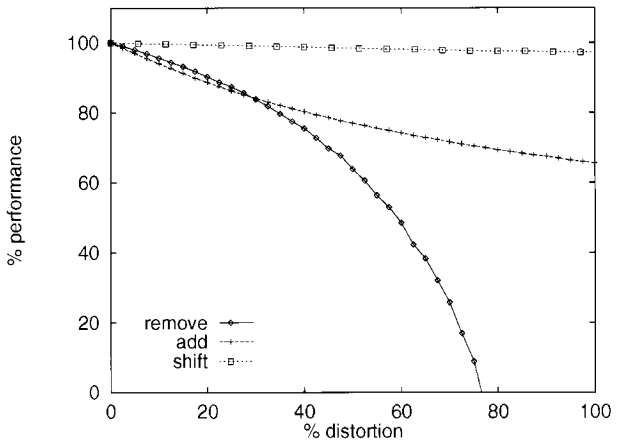

FIG. 5. Influence of removal, addition, and displacement of points on the scale normalization measure d for k = 10. Average percentage error over 25 trials.

Hippocampal CA Region Segmentation

The new method of distance graph matching was tested on the segmentation of the CA region in rat hippocampi (Fig. 6a), based on the preservation of the CA1 structure after ischemia (21). Here, the correlation between manual and automatic counting of the preserved cells in the CA1 region is shown. An example of cell detection is shown in Figure 6b. As a result from the distance graph matching, all cells in the CA and Hillus region were extracted from the image (Fig. 6c). Only cluster cells are preserved in the segmented image.

The CA1 region (Fig. 6) is that part inside the CA region, starting orthogonally at the end of the CA inside the hillus, and ending where the CA region becomes thicker before the U-turn. Manual counting was performed on 2–4 slices for each animal, resulting in a total number of preserved neurons counted in a total length of CA1 region (cells/mm) per animal.

To demonstrate the usefulness of the proposed segmentation method, correlation between these manual countings and automatic counting is shown. Due to the ambiguous definition of the CA1 region, manual indication of the CA1 region in the hippocampus image was necessary. For each hippocampus, three points were obtained, indicating the start (S), middle (M), and end (E) of the CA1 region. The segmented cells between start and end point, and within a reasonable distance from the line segment SME connecting the three points, were classified as belonging to the CA1 region. The average amount of cells per unit length was calculated for the obtained cell cluster. The cluster length was taken to be the length of line segment SME. Figure 7 shows the correlation between the manual and automatic counting for each of the algorithms tested. Results obtained with segmentation based on the Voronoï graph and for the nearest-neighbor distance do not corre-

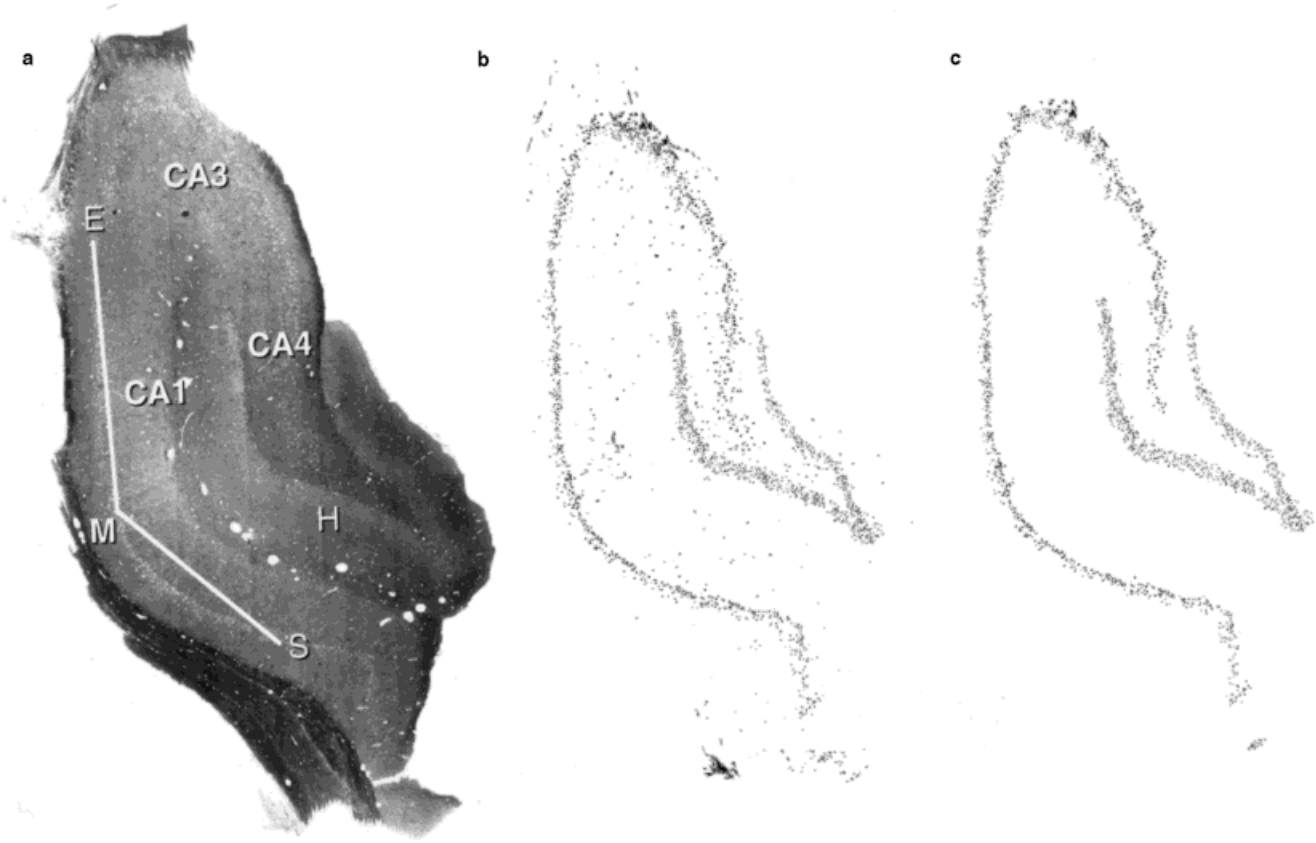

FIG. 6. Example of the segmentation of cell clusters in the hippocampus of a rat. The line segment SME indicates the CA1 region. All segmented cells in figure (c) between points S and E are considered part of the CA1 region. The length of the CA1 region is derived from the length of line SME. **a:** Hippocampus image as acquired by the setup. **b:** The resulting image from the cell detection. **c:** Cell clusters after the distance graph matching.

late well with manual counting. The method of Lavine et al. (14) is biased (mean error, −12.9) and results in a mean squared error of 405.0 (22). For the distance graph matching algorithm, the mean error is 0.1 and the mean squared error is 174.8.

DISCUSSION

The geometrical arrangement of cells in tissues may reveal differences between physiological and pathological conditions based on structure. This intuitive notion does not imply the quintessence that the arrangement can be captured easily in an algorithm. Quantification of tissue architecture, when successful and objectively measurable, opens the way to better assessment of treatment response. Before deriving parameters from tissue architecture, partitioning of the tissue in its parts of interest is necessary.

We present a method for the segmentation of homogeneous tissue parts based on cell clustering. The objective is to develop a method which is robust under spatial distortions intrinsic to the acquisition of biological preparations, such as squeezing the tissue as well as taking a two-dimensional transection through a three-dimensional block. These manipulation artifacts lead to two major confounding factors: 1) distortion in cell density, and 2) errors in cell detection. Distortion in cell density is reflected in the distance between cells. Irregularity or spatial distortion in the cell positions, and thus distortion in the neighbor distances, is inherent to tissues. Squeezing of tissue or local nonrigid deformations result in structural changes in cell density and thus changes in neighbor distances. Small changes in transection angle cause loss of cells in regions of the tissue. A second source of error in cell detection is the classification of artifacts as cells, or else cells may be overlooked during detection, causing lack of proper definition of local tissue architecture. When neighboring cells touch one another, they are often erroneously detected as one single cell. The method also deals with the uncertainty in cell classification often encountered in the automatic processing of tissues. Errors in the assignment of cells on cluster borders should be minimal to prevent influence of cluster shape on the segmentation result. The quantitative method enables reliable classification of areas by type of tissue.

In contrast to other cell pattern segmentation methods, the proposed distance graph matching algorithm meets the various demands as formulated above. Detection errors such as missing cells or artifact detection are corrected by respective insertion and deletion operations.

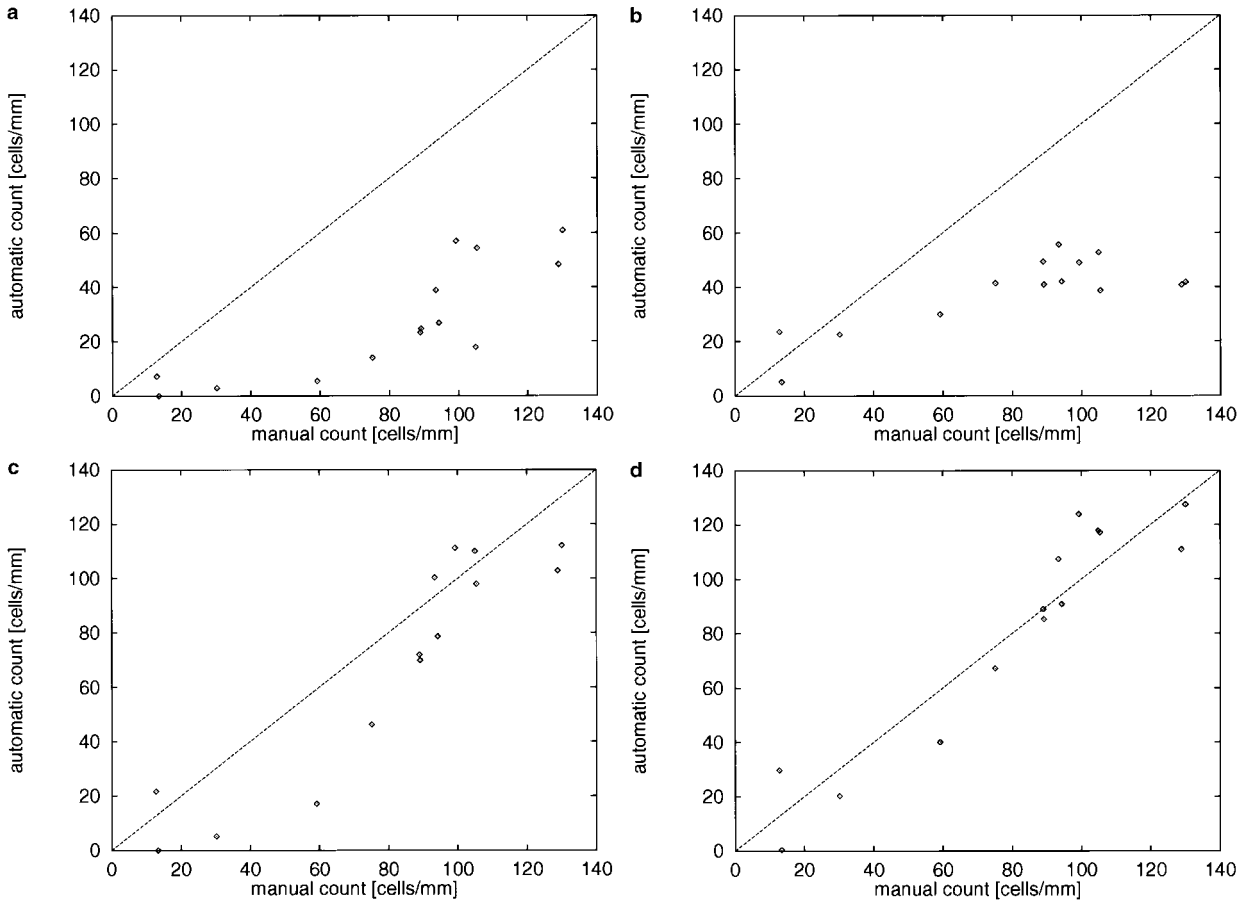

FIG. 7. Correlation between average number of cells per mm CA1 length per animal counted manually, and the number of segmented cells per estimated mm CA1 length per animal. Dashed line indicates $y = x$. **a:** Voronoï method. **b:** Nearest-neighbor method. **c:** Method of Lavine et al. (14). **d:** Distance graph matching method.

Deviation of the distances to neighboring cells is incorporated by allowing some tolerance in distance matching. Local deformation of the tissue has only minor influence as long as the deviation in distances remains within tolerance. The total sum of errors, combined with deviation in distances, indicates how well the cell and its environment fit the prototype environment. A possible drawback of the algorithm is its insensitivity to orientation. It is possible for two different patterns to have the same distance graphs. Under these circumstances, segmentation is not possible by any algorithm based on interpoint distances.

Including cell probability in the matching process further improves segmentation performance. The interference between the probability indicating that the object is or is not a cell, and the fit of the object in the cluster prototype, allow a better rejection of artifacts, while cluster cell classification is less affected. Cell confidence levels can be derived from the evaluation of the probability distribution of cell features as contour ratio. In order to remain independent of microscope and camera settings, the cell features chosen should not depend on scale, absolute intensity, etc.

The selection of an example often involves a supervised (i.e., interactive) procedure. The design of such a procedure requires adherence to several principles (23). Among other requirements, reproducibility under the same intention is considered the most important for our purpose. As a consequence, any prototype selection algorithm should only consider cells in conformity with the expert's intention.

Application of the method to the detection of the CA structure in rat hippocampi showed that even narrow elongated structures, only a few cells thick, can be well-segmented using the proposed distance graph matching. Results obtained semiautomatically correlate well with manual countings of preserved cells in the CA1 region, as long as there are enough cells left to discern regular clusters. The other segmentation methods tested, based on the area of influence in the Voronoï graph, the distance to the nearest neighbor, and the method of Lavine et al. (14), resulted in poor correlation between automatic segmentation and the countings by the expert. For the case of CA region determination, the proposed method proved to be compatible with the perception of the

pathologist. We have not applied the method to other tissue segmentation problems.

For the recognition of tissue architecture, the proposed distance graph matching algorithm has proven to be a useful tool. The method reduces the nonbiological variation in the analysis of tissue sections and thus improves confidence in the results. The present method can be applied to any field where regular patterns have to be recognized, as long as the directional distribution of neighbors may be neglected.

ACKNOWLEDGMENTS

The authors are grateful to Jos van Reempts and Marc Haseldonckx for the tissue preparations and for their advice; to Lambert Leijsen for the illustrations; and to Luc Wouters for comments on statistical evaluation.

LITERATURE CITED

1. Chandebois R. Cell sociology: a way of reconsidering the current concepts of morphogenesis. Acta Biotheor (Leiden) 1976;25:71–102.
2. Dormer KJ. Fundamental tissue geometry for biologists. London: Cambridge University Press; 1980. p 91–145.
3. Honda H. Geometrical models for cells in tissues. Int Rev Cytol 1983;81:191–248.
4. Bigras G, Marcelpoil R, Brambilla E, Brugal G. Cellular sociology applied to neuroendocrine tumors of the lung: quantitative model of neoplastic architecture. Cytometry 1996;24:74–82.
5. Guillaud M, Matthews JB, Harrison A, MacAulay C, Skov K. A novel image cytometry method for quantification of immunohistochemical staining of cytoplasmic antigens. Anal Cell Pathol 1997;14:87–99.
6. Raymond E, Raphael M, Grimaud M, Vincent L, Binet JL, Meyer F. Germinal center analysis with the tools of mathematical morphology on graphs. Cytometry 1993;14:848–861.
7. Darro F, Kruczynski A, Etievant C, Martinez J, Pasteels JL, Kiss R. Characterization of the differentiation of human colorectal cancer cell lines by means of Voronoï diagrams. Cytometry 1993;14:783–792.
8. Palmari J, Dussert C, Berthois Y, Penel C, Martin PM. Distribution of estrogen receptor heterogeneity in growing MCF-7 cells measured by quantitative microscopy. Cytometry 1997;27:26–35.
9. Rodenacker K, Bischoff P. Quantification of tissue sections: graph theory and topology as modelling tools. Pattern Rec Lett 1990;11:275–284.
10. Meijer GA, Beliën JAM, van Diest PJ, Baak JPA. Image analysis in clinical pathology. J Clin Pathol 1997;50:365–370.
11. Vincent L. Graphs and mathematical morphology. Signal Processing 1989;16:365–388.
12. Ahuja N, Tuceryan M. Extraction of early perceptual structure in dot patterns: Integrating region, boundary, and component gestalt. Comput Vision Graphics Image Process 1989;48:304–356.
13. Marcelpoil R, Usson Y. Methods for the study of cellular sociology: Voronoï diagrams and parametrization of the spatial relationships. J Theor Biol 1992;154:359–369.
14. Lavine D, Lambird BA, Kanal LN. Recognition of spatial point patterns. Pattern Rec 1983;16:289–295.
15. Rao CR, Suryawanshi S. Statistical analysis of shape of objects based on landmark data. Proc Natl Acad Sci USA 1996;93:12132–12136.
16. Schwarz H, Exner HE. The characterization of the arrangement of feature centroids in planes and volumes. J Microsc 1983;129:155–169.
17. Sankoff D, Kruskal JB. Time warps, string edits and macromolecules: the theory and practice of sequence comparison. Reading: Addison-Wesley; 1983. p 1–44.
18. Duyckaerts C, Godefroy G, Hauw JJ. Evaluation of neuronal numerical density by Dirichlet tessellation. J Neurosci Methods 1994;51:47–69.
19. Beucher S, Meyer F. The morphological approach to segmentation: the watershed transformation. In: Dougherty ER, editor. Mathematical morphology in image processing. New York: Marcel Dekker; 1993. p 433–481.
20. Wallet F, Dussert C. Multifactorial comparative study of spatial point pattern analysis methods. J Theor Biol 1997;187:437–447.
21. Haseldonckx M, Van Reempts J, Van de Ven M, Wouters L. Protection with lubeluzole against delayed ischemic brain damage in rats. Stroke 1997;28:428–432.
22. Sheiner LB, Beal SL. Some suggestions for measuring predictive performance. J Pharmacokinet Biopharm 1981;9:503–512.
23. Smeulders AWM, Olabariagga SD, van den Boomgaard R, Worring M. Design considerations for interactive segmentation. In: Jain R, Santini S, editors. Visual information systems 97. San Diego: Knowledge Systems Institute; 1997. p 5–12.

APPENDIX

The dynamic programming solution for matching the observation with the prototype is given in Figure 8. The graph searches a small set (horizontal) inside a larger set (vertical). The graph represents horizontally the prototype set $P = \{p_1, p_2, \ldots, p_k\}$ and vertically the input set $N_v^l = \{d_1, d_2, \ldots, d_l\}$. Each node C[i,j] in the graph represents the comparison between the i^{th} element from the prototype with the j^{th} element from the input set.

The directional edges in the graph determine which operations (deletion, insertion, or matching/substitution) are necessary to obtain the observed and prototype distance at the same position in the comparison string. For instance, each valid path from ''start'' to node C[2,3] describes the operations necessary to end up with a set where the third element in the observation is considered as the second one. A horizontal step represents insertion of the prototype element; the same observed distance is compared to the next prototype element. A vertical step implies deletion of an observation; the next observed distance is compared to the prototype element. Matching or substitution is represented by a diagonal step.

A cost is assigned to each edge. Using an edge to reach a particular node implies the addition of the edge penalty to the total cost involved for reaching the node. Horizontal edges have cost c_i; vertical edges cost c_d. The cost for diagonal edges depends on the comparison between the elements connected to the node from which the arrow starts. The cost is zero when the elements match ($c_m = 0$), c_s when the observed element is substituted for the prototype (when $c_s \leq c_m$), or cost c_m for making them match otherwise.

The cost to reach a particular node is the sum of all costs necessary when taking some valid path from ''start'' to the node considered. The minimum cost to reach that node is related to the path with the least total cost compared to all other possible paths. When considering only the previous nodes, i.e., all nodes from which the one under consideration can be reached, the problem can be reformulated into a recurrent relation. In this case, the minimum cost path is given by the least of the minimum cost paths to the previous nodes, increased by the cost for traveling to the node of interest.

Comparison begins at the ''start'' node, and each column is processed consecutively from top to bottom. In this manner, the minimum cost paths to the previous nodes are already determined when arriving at a particular node. The minimum cost to reach the node under consid-

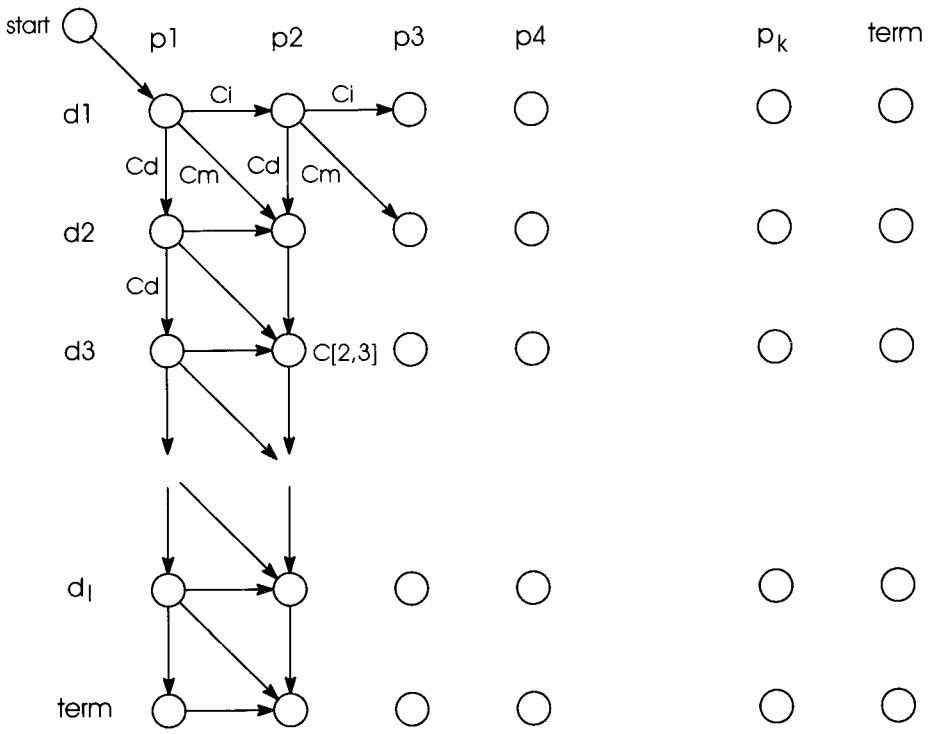

FIG. 8. The dynamic programming solution for string matching.

eration is then given by:

$$C[i, j] = \min \begin{cases} C[i, j-1] + c_d \\ C[i-1, j] + c_i \\ C[i-1, j-1] + c_m \\ C[i-1, j-1] + c_s \end{cases}. \quad (7)$$

The initial value at ''start'' is zero; the cost from ''start'' to the first node is also zero. The cost assigned to nonexisting edges (at the border of the graph) is considered infinity.

The ''term'' nodes at the bottom and right side of the graph are used for collecting the matching costs assigned to matching the last element in the observation (bottom) or the last element from the prototype (right). The term node C[k + 1, l + 1] describes the costs associated with matching the input set exactly to the prototype. The only interest is in finding the prototype in a (larger) number of observed distances, for which the cost is given by node C[k + 1, k + 1]. This is the first node where the observation is exactly transformed in the prototype. When there exist additional insert and delete operations on the observed set which result in a smaller matching cost, this path should be taken as the minimum cost path. Therefore, the minimum total cost is given by the minimum of the term nodes from C[term, k + 1] to C[term, l + 1].

The order of the string matching algorithm is O(l × k) (17). Here, k is the amount of neighbors in the prototype, and l is the amount of neighbors taken from the observation. When the cost for matching is constant, as is the cost for substitution, then algorithms with a lower complexity are known to compare ordered sequences.