CORRESPONDENCE ANALYSIS

Visualizing property-profiles of time dependent 3D datasets

Karsten Fries
University of Kaiserslautern, Department of Computer Science
D-67653 Kaiserslautern, Germany
kfries@informatik.uni-kl.de

Jörg Meyer
Mississippi State University, Department of Computer Science
Mississippi State, MS 39762
jmeyer@cs.msstate.edu
http://www.cs.msstate.edu/∼jmeyer

Hans Hagen
University of Kaiserslautern, Department of Computer Science
D-67653 Kaiserslautern, Germany
hagen@informatik.uni-kl.de
http://davinci.informatik.uni-kl.de

Bernd Lindemann
Saar University, Department of Physiology
D-66421 Homburg, Germany
phblin@med-rz.uni-saarland.de
http://www.med-rz.uni-sb.de/med_fak/physiol2/LDM/index.html

Keywords: correspondence analysis, tracking, matching, property-profile, minimum spanning tree, histogram, clusters

Abstract The article presents a method to perform an analysis of correspondence between sets of points in three-dimensional euclidean space E^3. Application-specific spatial data structures like the minimum (euclidean) spanning tree and several kinds of histograms assessing different transformations combined with quantities characterizing geometrical and topological qualities of point clusters are used to compute scores for point-to-point identification. These ratings are accumulated in a so-called match matrix, which is finally employed to extract a 1:1 match. The method is used to track individual fluorescent spots (synapses) in a volume of tissue which undergoes uneven spatial distortion (swelling and shrinkage). This enables the creation and analysis of cell property-profiles.

Introduction

In several areas of research a study of the internal processes of a system is only possible if objects of one state can be matched to objects of another state. The tracking of identical objects is necessary to determine their properties and their change in the course of time and/or treatment.

In biology and especially physiology often problems arise, where a matching algorithm (correspondence analysis) becomes essential. The tracking of cells or organisms that lack deterministic dynamics in terms of movement in a cellular sample is only one problem category out of a variety of similar applications [4].

The work presented in this paper originated in a biomedical application, i.e. the tracking of synaptical structures in taste buds of rats or frogs. The objects (fluorescent synapses) of one set of 3-D data are matched to those of a preceding set. Thus individual spots can be tracked despite uneven swelling and shrinkage of the tissue. This allows the eventual construction of intensity-time profiles of individual synapses.

1. NOTATIONS

In the following paragraphs the sets S (source) and D (destination) represent two sets of points in three-dimensional euclidean space E^3 formalizing object positions in the origin and destination state. $S = \{s_1, ..., s_n\}$, $D = \{d_1, ..., d_m\}$. The points in the sets are associated with intensities or rather weights for generality. They are denoted as $I(s)$, $s \in S$, as well as $I(d)$, $d \in D$. The corresponding position vectors are written as $\vec{s}$ and $\vec{d}$.

There are two different views on sets used in this context. Ordered sets contain implicit information about the topology of the addressed points and therefore define series of edges. In an unordered set, points are not connected in any specific way. Subsets of S are denoted by $P = (p_1, ..., p_k) \subset S$ in the ordered case and by $P = \{p_1, ..., p_k\} \subset S$ in the unordered case. Subsets of D are strictly named with $Q \subset D$ having cardinality $l = |Q|$.

2. BASICS AND DEFINITIONS

In order to formulate the matching algorithm in a single paragraph, the basics must be considered in advance. The following sections introduce the different components of the correspondence analysis.

2.1. EUCLIDEAN MINIMUM SPANNING TREE

Definition 1. *An undirected graph Γ consists of a set of nodes (vertices) V_Γ and a set of undirected edges $E_\Gamma \subset V_\Gamma \times V_\Gamma$.*

Remark 1. Since a graph Γ is undirected, $\{p_1, p_2\}$ and $\{p_2, p_1\}$, $p_1, p_2 \in V_\Gamma$, represent the same edge.

Definition 2. *Let M denote an unordered set of points in the three-dimensional euclidean space E^3. A graph Γ consisting of nodes that represent the points of the set $V_\Gamma = M$ with the properties*

- *Γ is connected (from each node $p \in V_\Gamma$ every further node $q \in V_\Gamma$ is reachable by a sequence of edges in E_Γ).*
- *The sum of the euclidean lengths of all edges of Γ*

  $$L(\Gamma) := \sum_{\{p,q\} \in E_\Gamma} ||\vec{q} - \vec{p}||_2 \qquad \textit{(1.1)}$$

  is minimized.

is called euclidean minimum spanning tree.

Detailed definitions and additional information can be found in special graph theory literature. In this context the *minimum spanning tree* (MST) is used to apply a structure to unordered points in E^3. It has useful properties like connecting clusters of points that potentially reoccur in the second dataset, and it is unambiguous if all point-to-point distances are different in pairs. Moreover, it reduces edge count complexity to $\mathcal{O}(n)$ and $\mathcal{O}(m)$ in the two denoted sets, instead of $\mathcal{O}(n^2)$ and $\mathcal{O}(m^2)$ for the case that all points are connected to each other (fully connected set). Unfortunately the *minimum euclidean spanning trees* of S and D can differ because of small variations in the point sets (even if S and D have equal cardinality). Therefore one should add a so-called *closure* to the MST Γ_D.

Definition 3. *Let Γ_M be the minimum euclidean spanning tree of a set $M \subset E^3$, $2 \leq s \in N$. An ordered subset $P = (p_1, ..., p_s) \subset M$ with*

$\{p_1, p_2\}, ..., \{p_{s-1}, p_s\} \in E_{\Gamma_M}$ is called sequence or s-sequence in Γ_M. We define $\Lambda_s^{\Gamma_M}$ as the collection of all s-sequences of Γ_M. Moreover we represent sets of specific sequences ($t, s_1, ..., s_t \in N$) by

$$\Lambda_{s_1,...,s_t}^{\Gamma_M} := \bigcup_{i=1}^{t} \Lambda_{s_i}^{\Gamma_M}. \tag{1.2}$$

Definition 4. *Let $P \subset M$ denote a c-sequence ($3 \leq c \in N$) of a given graph Γ_M. A c-closure of P is an edge $\{p_1, p_c\}$ connecting the first and the last point of the the sequence P. For a collection of sequences $\Lambda_{s_1,...,s_t}^{\Gamma_M}$, we define $\Delta(\Lambda_{s_1,...,s_t}^{\Gamma_M})$ as set of closure edges of all sequences of $\Lambda_{s_1,...,s_t}^{\Gamma_M}$.*

Remark 2. Note that a *c-closure* always bridges $c - 2$ points.

In the following we address two sets of *sequences*: $\Lambda_{2,...,s_{max}}^{\Gamma_S}$ (denoted by Λ_S) and $\Lambda_{2,...,s_{max}}^{\Gamma_D{}'}$ (denoted by Λ_D), where $\Gamma_D{}'$ is Γ_D with extended edge set $E'_{\Gamma_D} = E_{\Gamma_D} \cup \Delta(\Lambda_{3,...,c_{max}}^{\Gamma_D})$. The constants s_{max} and c_{max} restrict the extracted *sequence* lengths. They are user-defined and we will have a closer look at them in terms of complexity.

To ease comprehensibility we rephrase our problem: The task is matching two sets S and D of objects that are represented by positions in euclidean space E^3. The cardinality of S and D may be unequal, due to objects disappearing or getting out of scope from one state to the other. We try to apply a structure to each set by constructing the MSTs Γ_S and Γ_D. In general the graphs differ, since the objects where exposed to various modifications. To compensate this difference, we extend Γ_D by adding further edges from *closures* of specified *sequences* from it and obtain Γ'_D. At last, to accomplish a comparisson between the sets, we extract *sequences* of certain lengths from Γ_S and Γ'_D, subsequently denoted by Λ_S and Λ_D.

Remark 3. Note that the extended graph Γ'_D is not fullfilling *minimum spanning tree* properties anymore.

2.2. CHARACTERISTIC VALUES

In order to compare unordered sets of points, it is necessary to define criteria or quantities to describe an identity score or a similarity error, respectively. This paragraph introduces and discusses the characteristic values proposed.

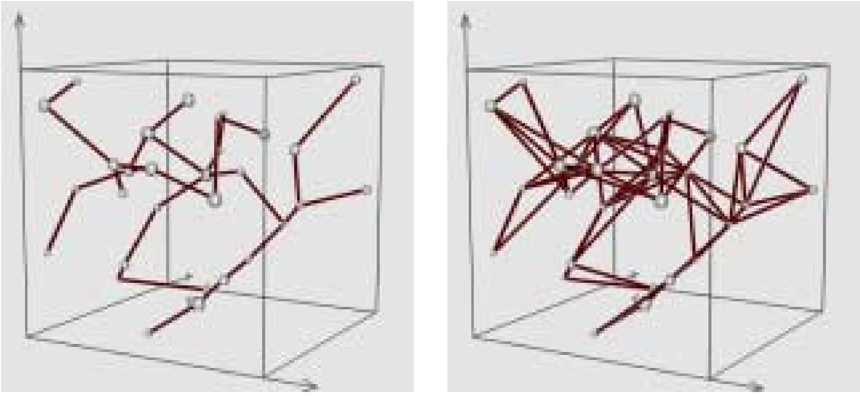

Figure 1 *Minimum euclidean spanning tree* in three-dimensional space. Right: the 3-*closure* supplements the graph with further edges.

2.2.1 Accounting for Relative Position. To account for the relative position of the sets elements the centroid $\vec{c}(P)$ is computed in advance and the euclidean distances from it are summed up.

$$\vec{c}(P) = \frac{1}{|P|} \sum_{p \in P} \vec{p} \tag{1.3}$$

$$cv_1(P) = \frac{1}{|P|} \sum_{p \in P} ||\vec{c}(P) - \vec{p}||_2 \tag{1.4}$$

This characteristic value is linear with respect to scaling: $cv_1(\lambda P) = \lambda cv_1(P)$, $\lambda \in \mathbf{R}_\mathbf{0}^+$, and invariant under rigid motions $\mathcal{A}$: $cv_1(\mathcal{A}P) = cv_1(P)$.

2.2.2 Accounting for Edge Length. Due to the missing order we must assume that the set is fully connected. This means each pair of points defines an edge.

$$cv_2(P) = \frac{1}{\binom{|P|}{2}} \sum_{\substack{p_i, p_j \in P \\ i<j}} ||\vec{p_j} - \vec{p_i}||_2 \tag{1.5}$$

This second characteristic value has the same mathematical properties as cv_1.

2.2.3 Accounting for Object Intensity. As long as the intensity is not the interesting value to be monitored by the analysis, it is possible to use it in a third characterizing quantity. For example a simple averaging can be done.

$$cv_3(P) = \frac{1}{|P|} \sum_{p \in P} I(p) \tag{1.6}$$

The properties of this value depend on the qualities of the value $I(p)$ of $p \in P$.

Remark 4. The normalizing factors are meant to obtain values of the same order of magnitude. We assume the object intensities being adapted to these requirements.

2.2.4 Combining Characteristic Values. At last, these characteristic values are combined to an already mentioned similarity error of two sets P and Q.

$$\varepsilon(P, Q) = \sum_{i=1}^{3} c_i |cv_i(Q) - cv_i(P)| \tag{1.7}$$

The mathematical quality of the error depends on the properties of the quantities used. Assuming that only the first two characteristic values defined in the previous sections are combined, the error also shows linearity and invariation under rigid motions, such as translations and rotations. The constants c_i rate the respective values. The similarity error can be used to define an identity score

$$\frac{1}{1 + \varepsilon(P, Q)}. \tag{1.8}$$

As a result the formula yields 1 for equality of the sets and converges to 0 with more and more dissimilar sets P and Q.

2.3. USING HISTOGRAMS

To obtain structural, geometric and topological information from S and D, two different histograms are built upon series of edges from Γ_S and the supplemented Γ_D. The matching algorithm can use the data from the histograms to compose the scores that will be described subsequently. To be able to compare and to use the histograms in the scoring procedure, they will be normalized denoting $\hat{H}$.

2.3.1 Translation Histogram. For the generation of the histograms the underlying space (three-dimensional space in case of translation in E^3) is divided into cells that accumulate scores for a respective group of translations. The score for the correspondence between two

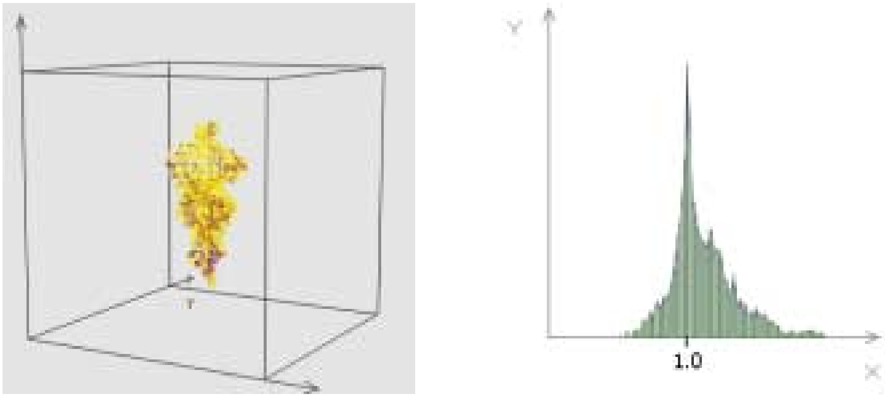

Figure 2 Translation histogram of a spiral dataset (left) and a scaling histogram with strongly developed rating for $\lambda = 1$ in its center (right).

sequences P and Q, obtained from the combination of the characteristic values in $\varepsilon(P,Q)$, is added as described by the following formula.

$$H^{(r+1)}_{translate}(\vec{v}) = H^{(r)}_{translate}(\vec{v}) + \frac{1}{1+\varepsilon(P,Q)} \tag{1.9}$$

The vector $\vec{v} := \vec{c}(Q) - \vec{c}(P)$ defines the cell to which the score is added. Thus, in case of D being created by simple translation of S, $H_{translate}(\vec{c}(D) - \vec{c}(S))$ will receive the highest score.

2.3.2 Scaling Histogram. In analogy to the translation histogram the scores are added to the scaling histogram. The underlying space is one-dimensional in this case and divided into intervals.

$$H^{(r+1)}_{scale}(\lambda) = H^{(r)}_{scale}(\lambda) + \frac{1}{1+\varepsilon(P,Q)} \tag{1.10}$$

The formula determines the scoring procedure for each pair of sets P and Q. The interval identifying factor $\lambda \in \mathbf{R}^{+}_{\mathbf{0}}$ is computed by $\lambda := \frac{cv_1(Q)}{cv_1(P)} = \frac{cv_2(Q)}{cv_2(P)}$, for $cv_1(P) \neq 0$ and $cv_2(P) \neq 0$.

2.4. ACCUMULATION OF SCORES IN THE MATCH MATRIX

In the following paragraphs the composed votes are accumulated in a structure called *match matrix* or *vote scheme*. It can be seen as a special histogram structure and is therefore expressed in the same way as the translation and scaling histograms discussed above. The n rows of the

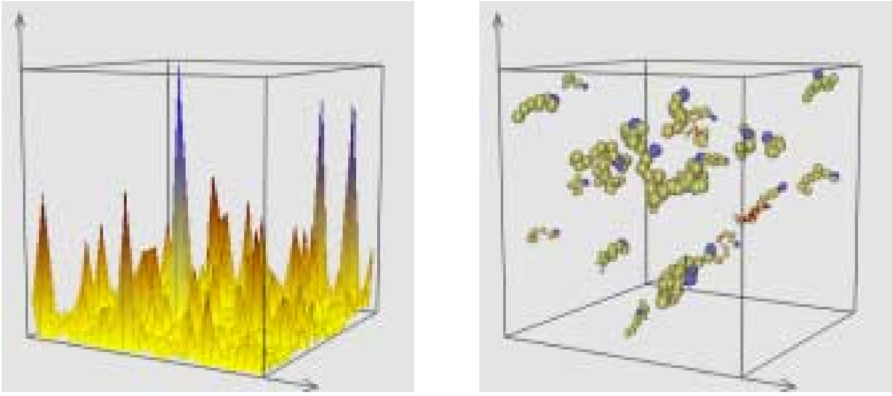

Figure 3 Example of a correspondence analysis vote scheme (left) and a result of matching multiple datasets (right). The blue (dull) spots identify the points from the source and the yellow (brighter) ones those of the destination set.

matrix H_{match} are associated with the points of the source set S and its m columns can be identified with the objects of the destination set D. This means, after the scoring procedure with kernel

$$H_{match}^{(r+1)}(i,j) = H_{match}^{(r)}(i,j) + v(i,j) \tag{1.11}$$

where $v(i,j)$ is a supplementing score, $H_{match}(i,j)$ contains a rating for the correspondence of $s_i \in S$ to $d_j \in D$.

2.5. PROCESSING THE MATCH MATRIX

After all scores are summed up in the *match matrix*, the matrix can be processed in a specific, application-dependent way.

A first approach to extract a 1:1 match starts an iterated procedure that selects the maximum value in the matrix, e.g. $H_{match}(i,j), i \in \{1,...,n\}$ and $j \in \{1,...,m\}$, marks d_j as a matching point to s_i and clears the according row and column of H_{match} in order to avoid further matchings.

This apparent method is somehow greedy, because it always selects the highest entry, and we must be aware of the fact that it does not necessarily realize the optimum 1:1 match, which is defined by the maximized sum of the chosen entries of H_{match}. However, our results have shown that this method provides a fast and efficient technique.

2.6. SCORING OF SUBSET CORRESPONDENCE

With data from the histogram structure and the geometry, topology and attribute characterizing similarity error $\varepsilon(P,Q)$, a vote for the correspondence of two unordered sets P and Q can be computed. We propose the following formula.

$$v(P,Q) = \frac{c_4\hat{H}_{translate} + c_5\hat{H}_{scale} + c_6}{(c_4 + c_5 + c_6) + \varepsilon(P,Q)} \tag{1.12}$$

The constants c_4 and c_5 are weights rating the information from the translation and scaling histograms. The third factor is computed by $c_6 = 1 \Leftrightarrow (c_4 = c_5 = 0)$ or it is set to zero if this condition is not met. Thus it ensures a vote unequal to zero if the histogram data is not considered.

Going with this formula, we are now able to rate the similarity of two sets, but are still not aware of point-to-point scores. There are several possibilities to accomplish this submatching task. The next section introduces an efficient and very convincing approach.

2.7. SCORING OF POINT CORRESPONDENCE

For the rating of point-to-point correspondence, ordered sets of points are taken into account. These implicitly define series of edges having a cardinality of at least 2. Despite the ordered set view, the rating from equation (1.12) is computed in advance. Comparing sets of cardinality 2, the result of

$$v_{P,Q}(i,j) = v(P,Q) \tag{1.13}$$

is used as a vote for point-to-point scoring for $s_i \in P \subset S$ to $d_j \in Q \subset D$ and added to the *match matrix* as described in 2.4.

For larger sets of cardinality $|P| = |Q| > 2$, a temporary copy of the current *match matrix* is used. After the completion of edge matching the *match matrix* already contains valuable data that represents a full, but possibly unsecure match. This means the *vote scheme* $H_{temporary} = H_{match}$ is used to perfom the submatch task.

$$v_{P,Q}(i,j) = v(P,Q)\hat{H}_{temporary}(i,j) \tag{1.14}$$

Moreover, the $H_{temporary}$ matrix can be updated after each processing of a determined subset size. This will result in more and more precise submatches.

3. MATCHING ALGORITHM

This section combines the structures and methods summarized in paragraph 2. The introduced algorithm has been implemented in a software package which enables the analysis of datasets from statistical experiments in biological and neurophysiological applications [3], [7], [8].

3.1. PREPROCESSING THE INPUT SETS

The first step of the matching procedure is the preprocessing of the two input sets S and D.

3.1.1 Matching the Centroid. The two sets might have been exposed to a global transformation, and therefore the centroids of the whole sets, computed by $\vec{c}(S)$ and $\vec{c}(D)$, are brought together by moving all objects of D by the vector $\vec{v} = \vec{c}(D) - \vec{c}(S)$.

This first preprocessing step will of course affect the translation histogram. The ratings should now be centered in $H_{translate}$.

3.1.2 Extracting Ordered Sets. For both sets the *minimum spanning trees* Γ_S and Γ_D are built. To reduce single-point importance the *closures* of Γ_D are computed. The extraction of ordered subsets is done by simply extracting *sequences* of specified lengths from the graphs. The structural and geometric information of Γ_S and Γ'_D is inherited implicitly to the *sequences*. The sets of *sequences* of Γ_S are denoted by Λ_S and those of Γ'_D by Λ_D.

3.1.3 Generating Histograms. The creation of the histograms can be seen as prematching all *sequences*. All translations and scalings in a pair (P, Q), $P \in \Lambda_S$, $Q \in \Lambda_D$, $|P| = |Q|$ of two *sequences* receive a score in the appropriate structure. This means that the histograms are supplemented by the rating procedure in equations (1.9) and (1.10) in paragraph 2.3. After this, one has the opportunity to ask for probabilites of translations and scalings by a certain vector or a specific scaling factor. This information was not accessible *a priori*.

3.2. MATCHING THE INPUT SETS

After preprocessing is completed, we have translation and scaling histograms as well as the extracted *sequences* at our disposal. With this information we are able to compute a score for each pair of *sequences* from Λ_S and Λ_D with equation (1.12) and add the rating to the *match matrix* as proposed in paragraph 2.7. As a last step, the processing of

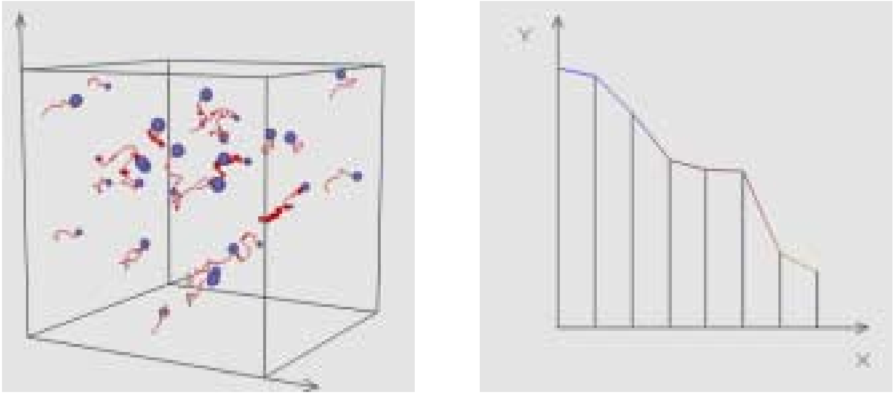

Figure 4 Representation of match results without destination objects (left). Single object intensity profile (right).

the *match matrix* is performed. An application-dependent method of the extraction of a 1:1 match was described in paragraph 2.5.

3.3. COMPLEXITY

In the implemented application the user needs to adjust the parameters according to the demands of the current task. Two very useful values are given by the *max. sequence length* s_{max} and the *max. closure length* c_{max} variables. The first defines the maximum length of *sequences* that are extracted from Γ_S and Γ'_D, and the second limits the *closures* that are used to supplement the graph of Γ_D.

Regarding complexity, the number of *sequences* strongly depends on these two values. In the paragraphs below they are denoted by s and c. At first we will discuss the simple comparison of edges from the pure *minimum spanning trees* (no supplement by *closures*). For the completion by *closures*, we found two approximative but expressive estimates.

3.3.1 Counting Sequences in the Minimum Spanning Trees.

The *minimum spanning tree* of a set with size n cannot differ in the number of edges (2-*sequences*), but in counts of s-*sequences*, where $n \gg s > 2$. We found the estimate for a set M

$$(n - s + 1) \leq E_s(M) \leq \delta_s^{max}(n - s + 1), \tag{1.15}$$

where $E_s(M)$ is the number of s-*sequences* in Γ_S, and $\delta_s^{max} = \frac{1}{2}(n - s)$ is a worst case factor. Further we conclude

$$E_s(M) \leq \delta_s(n - s + 1), \tag{1.16}$$

with δ_s being a *minimum spanning tree* dependent value limited by $1 \leq \delta_s \leq \delta_s^{max}$.

Comparing the resulting *sequence* sets resulting from the input sets S and D, we obtain $\Theta(\delta_s(S)(n-s+1)\delta_s(D)(m-s+1))$, and $\Omega(nms+s^2)$, $\mathcal{O}(n^2m^2s^2+s^4)$, respectively. Using constant input sets and simply varying s, we establish $\Omega(s^2)$ and $\mathcal{O}(s^4)$.

3.3.2 Effect of Closures on Complexity. As mentioned before, we will limit the analysis to two estimates. The first will be made based on a *c-closure* of Γ_D. An observation of the *closure* effect establishes the following fact: *'every c-closure yields a supplement of at least one further s-sequences in s respective nodes'*, and with the knowledge from the previous section ($E_c(D) \geq m-c+1$), we obtain $E_c'(D) \geq E_c(D) + s(m-c+1)$. The estimate for the lower bounds then reads

$$\Omega(E_c(S)E_c'(D)) = \Omega(nms + s^2c). \qquad (1.17)$$

This shows a mutual dependency on s and c that causes a higher complexity for the comparison of the sets.

The second case aims at an estimate for the upper complexity bound, which proves to be independent of c. We need to be aware of the fact that the worst case setting, the fully connected set, can occur with only a single *closure*. Due to this unfortunate effect we have to consider a complexity of

$$\mathcal{O}(\binom{n-s+1}{2}\binom{k!}{(k-m)!}) = \mathcal{O}(n^2m^s + s^2m^s). \qquad (1.18)$$

In this context a confrontation of the determined bounds with the complexity of the comparison of two worst case sets again justifies the use of the *minimum spanning tree* ($\mathcal{O}(|M|^3)$ for an unpreprocessed set M). To match the fully connected sets S and D by using *s-sequences*, we have to deal with

$$\Theta(n^sm^s) \qquad (1.19)$$

score computations.

4. CONCLUSION AND FUTURE WORK

The presented method can be utilized to analyse two sets of points in the three-dimensional euclidean space in order to find a satisfying match after an exposure to various modifications. For our application, the matching of corresponding synapses, we found that the introduced method provides high accuracy, which is manifested in the rigid movement of clusters that were identified by the geometric analysis. The tests refering to the histograms, showed very encouraging results and proved their indispensability. The future work will focus on considering an extraction of clusters from the histograms in order to reduce the existing noise caused by definitely uncorresponding objects.

In the previously mentioned application, we implemented another option supplementing the structure of the *minimum spanning tree*. The proposed method handles some sort of a hierarchical extension. A classification on aspects of dynamics or reliability is used to create *ranking sets* that are matched with respect to those of the destination set. However, this is only possible if the classification supports the matching. This means that objects of lowest dynamics in S will be matched with high probability to objects of low dynamics in D and so on. More details on the implemented algorithms, results, a discussion and further comments illustrating the methods used can be found in [3].

This work was supported by the Deutsche Forschungsgemeinschaft, SFB 530/B2.

References

[1] Alt H., Guibas L.J.: *Matching, Interpolation, and Approximation. A Survey.* TR B 96-11, Institut für Informatik, University of Berlin (1997)

[2] Foley T., Hagen H., Nielson G.: *Visualizing and Modelling Unstructured Data.* Proceedings GMD-Conference ”Visualisierung” (1992)

[3] Fries, K.: *Visualisierung und statistische Analyse synaptischer Strukturen.* Diploma Thesis. Department of Computer Science, University of Kaiserslautern (1999)

[4] Hagen H., Schreiber, T.: *Scattered Data Algorithmen zur Umweltdaten-Visualisierung.* In: Denzer, R., Hagen, H., Kutschke, H.J. (eds.) Visualisierung von Umweltdaten: 22–28 (1990)

[5] Hoffmann F., Kriegel K., Wenk C.: *Matching 2D Patterns of Protein Spots.* TR B 97-13, Institut für Informatik, University of Berlin (1997)

[6] Huttenlocher D.P., Klanderman G.A., Rucklidge W.J.: *Comparing Images using the Hausdorff Distance.* IEEE Transaction on Pattern Analysis and Machine Intelligence. Vol. 15, No.9, 850-863 (1993)

[7] Lindemann B.: *Taste reception.* Physiological Reviews 76:719-766 (1996)

[8] Lindemann B.: *The Taste of sweet and bitter.* Current Biology 6(10): 1224-1237 (1996)

[9] Meyer, J., Hagen, H., Lohr, C., Deitmer, J.W.: *Interactive Navigation through Glial Cells.* Computer Graphics International (CGI '98), Minisymposium on "Scientific Visualization", Hannover, Germany: 73–77 (1998)

[10] Olivio J.C., Kahn E., Halpern S., Fragu P.: *Image Registration and Distortion correction in ion microscopy*, Journal of Microscopy, Vol. 164, 263-272 (1991)

[11] Weber G.: *Point Pattern Matching.* TR B 95-19, Institut für Informatik, University of Berlin (1995)