Text mapping: Visualising unstructured, structured, and time-based text collections

Vedran Sabol[a], Keith Andrews[b,*], Wolfgang Kienreich[a] and Michael Granitzer[a]
[a]*Know-Center Graz, Austria*
[b]*Graz University of Technology, Austria*

Abstract. Large collections of text documents are increasingly common, both in business and personal information environments. Tools from the field of information visualisation are being used to help users make sense of and extract useful knowledge from such collections.

Flat text collections are often visualised using distance calculations between documents and subsequent (distance-preserving) projection. Distance calculations are often based on a vector space of term vectors. Projection is often achieved with a force-directed placement algorithm.

Where extra information about a text collection is available, such as a topical hierarchy or some chronological ordering, it can be used to improve a visualisation. This paper gives an overview of text mapping techniques.

Keywords: Information visualisation, text mapping, term vectors, force-directed placement, hierarchy, structure, chronological

1. Introduction

Large collections of text documents are increasingly common, both in business and in personal information environments. In order to make sense of and extract useful knowledge from such collections, tools from the field of information visualisation [1,21] are increasingly finding application.

The idea behind information visualisation is to utilise the remarkable capabilities of the human visual perception system to rapidly and automatically detect patterns and changes in visual displays. This is know as *pre-attentive processing* and requires minimal cognitive effort [23].

A typical collection ot text documents, say 100 PDF files containing papers from a conference, is usually “flat”. That is to say, the collection does not have any structural information, such as a topical hierarchy or associative links between related papers. Where the text documents in a collection are already organised hierarchically or a hierarchical structure can be induced from the documents’ contents, the visualisation process can make use of the hierarchy to produce a better visualisation. Sometimes, a text collection is temporal in nature, such as a collection of newspaper stories or press releases. Such temporal information can also be utilised to create or improve visualisations.

2. Visualising unstructured collections

The basic steps in the traditional text visualisation pipeline are:

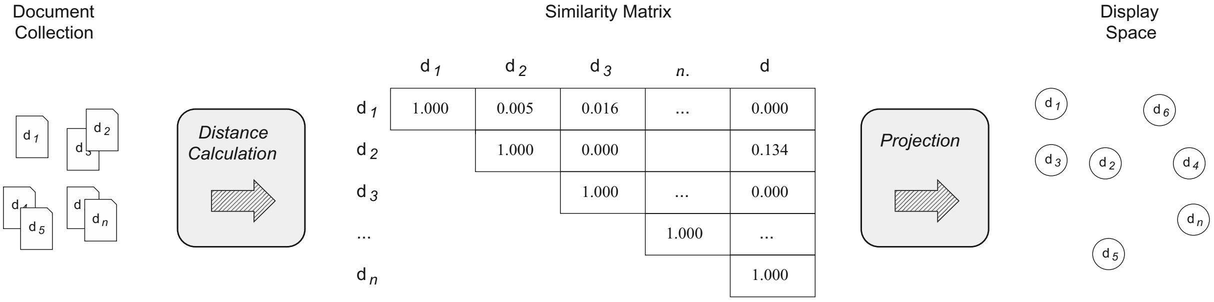
1. *Distance Calculation*: Calculate (dis)similarity values between every pair of objects (text documents).
2. *Projection*: Use the (dis)similarity values to place objects in a display space (usually a 2d or 3d space).

These steps are illustrated in Fig. 1.

In terms of distance calculation, the field of information retrieval (IR) has long used the *vector space model* to characterise the relationships between text documents [19]. Each text document is represented

[*] Corresponding author. E-mail: kandrews@iicm.edu.

ISSN 1872-4981/08/$17.00 © 2008 – IOS Press and the authors. All rights reserved



	d 1	d 2	d 3	n.	d
d 1	1.000	0.005	0.016	...	0.000
d 2		1.000	0.000		0.134
d 3			1.000	...	0.000
...				1.000	...
d n					1.000

Fig. 1. The traditional text visualisation pipeline.

by a *term vector* of high dimension (often many thousands). The number of dimensions is given by the total number of different words appearing in the text collection. Each entry in a document's term vector describes the frequency (or sometimes the significance) of a particular word in the document. The classic metric for distance between text documents is the cosine metric (scalar product), which expresses the angle between two vectors, and delivers values between 0 (completely dissimilar) and 1 (completely similar).

Eades [8] first proposed the use of forces between objects and iterative solution of an energy equation in his *spring model*. The original technique was later refined by Fruchtermann and Reingold [9] and given the name *force-directed placement* (FDP). The basic idea is that forces of attraction pull similar objects closer, while objects which are too close repel slightly.

Chalmers and Chitson [5] combined distance calculations derived from a vector space model with projection using force-directed placement, in a scheme similar to that shown in Fig. 2. In the distance calculation step, a term vector is first calculated for each document in the collection. The similarity between two documents is approximated by the scalar product (cosine metric) of the two term vectors.

To generate a landscape visualisation from the similarity matrix, a force model is constructed. For example, the sum force F_i acting on a particular document d_i might be given by the sum of attraction forces, repulsion forces, and a low-level gravitational force G:

$$F_i = \sum_{j \neq i} (att_{i,j} + rep_{i,j}) + G$$

The forces acting on document d_i are illustrated in Fig. 3. The force of attraction $att_{i,j}$ between two documents is proportional to the similarity between the two documents. The force of repulsion $rep_{i,j}$ between two documents is inversely proportional to the square of the distance between two documents in the display space (so that very close objects repel strongly). The force of repulsion should only kick in at very close inter-object distances. Often, a low-level gravitational force G is also used to draw all objects toward the centre of the display space and stop border objects drifting away.

Objects are initially placed randomly in the 2d display space, as illustrated in Fig. 4a. At each iteration, the forces acting on each object are calculated, and the object is moved a small amount in the direction of the total force F_i. After many (typically a few dozen) iterations, a final stable layout is reached, as shown in Fig. 4b.

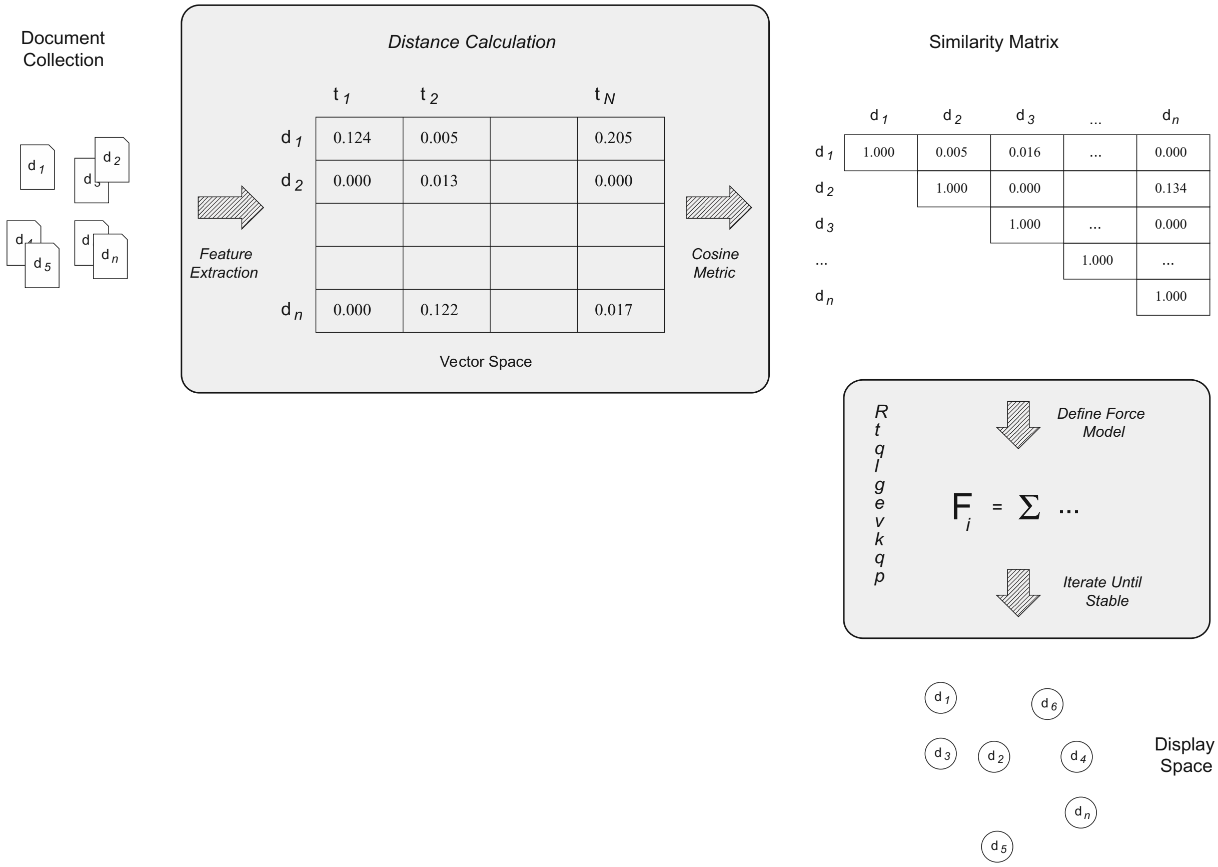
A brute force implementation of FDP would require the calculation of the forces between every pair of objects, resulting in a time complexity of $O(n^2)$ for each iteration. Chalmers and Chitson [5] used a heuristic for sampling a subset of objects to reduce this complexity to $O(n)$ for each iteration. Jourdan and Melancon [11] later further improved this $O(\log(n))$ for each iteration.

The SPIRE [13,24] and VxInsight [6] systems use a more intricate method of projection known as multidimensional scaling to project directly from the high-dimensional vector space to a 2d display space. Figure 5 shows VxInsight displaying a set of articles from physics journals. SPIRE's ThemeView introduced the notion of a topical hillscape, where groupings of the same terms "pile up" to form peaks in the landscape.

In order to explore the field of text mapping, there is a freely available tool called PEx (Project Explorer) [15], which allows various combinations of distance calculation and projection techniques to be experimented with.

3. Visualising structured collections

In cases where a collection of text documents features explicit hierarchical structure, or where a hierarchical structure can be deduced by analysing the col-



	t 1	t 2		t N
d 1	0.124	0.005		0.205
d 2	0.000	0.013		0.000
d n	0.000	0.122		0.017

	d 1	d 2	d 3	...	d n
d 1	1.000	0.005	0.016	...	0.000
d 2		1.000	0.000		0.134
d 3			1.000	...	0.000
...				1.000	...
d n					1.000

Fig. 2. Using a vector space for distance calculation and force-directed placement (FDP) for projection.

lection, improvements can be made to the “flat” techniques described in Section 2. The TreeMap [20] is an example of a space-constrained visualisation of hierarchical structures, where levels of the hierarchy are represented by nested rectangles. The size and the colour of these rectangles are used to encode additional information. Figure 6 shows a TreeMap visualising a directory structure. Each rectangle represents one hierarchy branch. The area of a rectangle corresponds to the total size (or somtimes number) of files contained in the represented branch, while colour is used to encode the type of the files (such as text, images, PDFs, etc.). The TreeMap representation has become quite popular and has been applied to a variety of areas, such as stock markets and news articles.

TreeMaps directly represent hierarchical relations as specified in the underlying structure; They are ill-equipped to convey other types of relations, for example topical similarity between collections or documents. Various approaches have been proposed to overcome this limitation. InfoSky [3] is an example of a system designed for visualising both explicitly defined hierarchical structure and topical similarity present in document and collection content. The system enables exploration of very large, hierarchically structured document repositories by employing the night sky as a metaphor: Documents are visualized as stars, while collections are visualized as polygonal shapes bounding groups of stars, resembling constellations in the night sky. As in TreeMaps, the area assigned to a collection corresponds to the total size (or number) of documents and sub-collections contained within that collection. Fur-

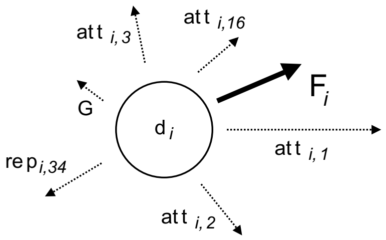
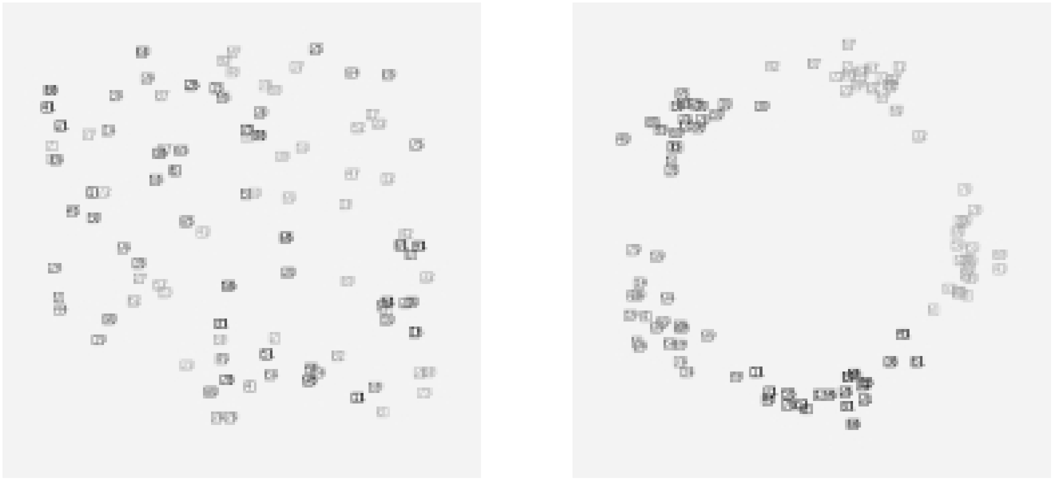


Fig. 3. The sum force F_i acting on document d_i is given by the vector sum of all forces acting on d_i.



(a) The initial state for force-directed placement. A set of objects is randomly placed in the 2d display space.

(b) After many iterations of force-directed placement, similar objects have moved toward each other to form visual clusters.

Fig. 4. Force-directed placement (FDP). Over many iterations, similar objects move closer together and dissimilar objects move further apart. (a) shows the initial state, (b) shows the final state.

thermore, collections and documents are positioned in the 2D space in a way that translates topical similarity into spatial proximity. Collections and documents are labelled with their titles (if available) as well as with automatically extracted keywords providing appropriate descriptions at any chosen level of magnification. Smooth, animated transitions allow the user to point the virtual telescope to a specific region (panning) and to vary the magnification level (zooming).

InfoSky builds upon the techniques described in Section 2 and additionally exploits the hierarchical structure to reduce computational complexity. Starting with an arbitrary polygonal area representing the repository as a whole, the following strategy is applied recursively from the root collection. Collection centroids, vectors representing collections, are computed as a sum of vectors belonging to documents contained by the collections. Corresponding similarity matrices are computed for all direct child collections. Then, a force-directed placement algorithm creates a two-dimensional layout for centroids, grouping similar collections (centroids) together and reserving more space around larger collections whenever possible. The layout results are inscribed into the polygonal area assigned to the collection currently being processed. Finally, polygonal areas are assigned to each child collection by computing a weighted Voronoi diagram from placed centroids and clipping it against current bounds. Weighting adjusts the edges of created polygons to assign larger areas to collections containing more documents. In collections which do not contain any further subcollections (leaves), documents are assigned locations using force-directed placement. Collections which contain both subcollections and documents have an artificial area reserved within them into which documents are placed.

This recursive procedure saves huge amounts of main memory space and processing time. The force-directed placement algorithm, which has quadratic time and space requirements, does not have to process the entire collection at once, but rather is only applied to the direct children of a single collection at a time. Provided no single collection has a very large number of direct children, the procedure allows for very fast processing of huge document collections. Furthermore, there is no need to keep large amounts of data in main memory at one time. The data set can be stored on disk and only the data for direct children of the currently processed collection need be fetched into main memory for processing at any one time.

Figure 7 shows the InfoSky browser visualising a repository of over 100,000 technical documents organised into a manually edited hierarchy up to 16 levels deep. By colour-coding search results (green is 'windows', magenta is 'linux') one can easily compare the distribution of results over the whole hierarchy and identify correlations. Users can pan and zoom freely or navigate into the hierarchy by clicking on collection labels. Figure 8 shows six levels of zooming around the collection "Webalizer".

As discussed above, where document collections are already pre-structured into a topical hierarchy, exploit-

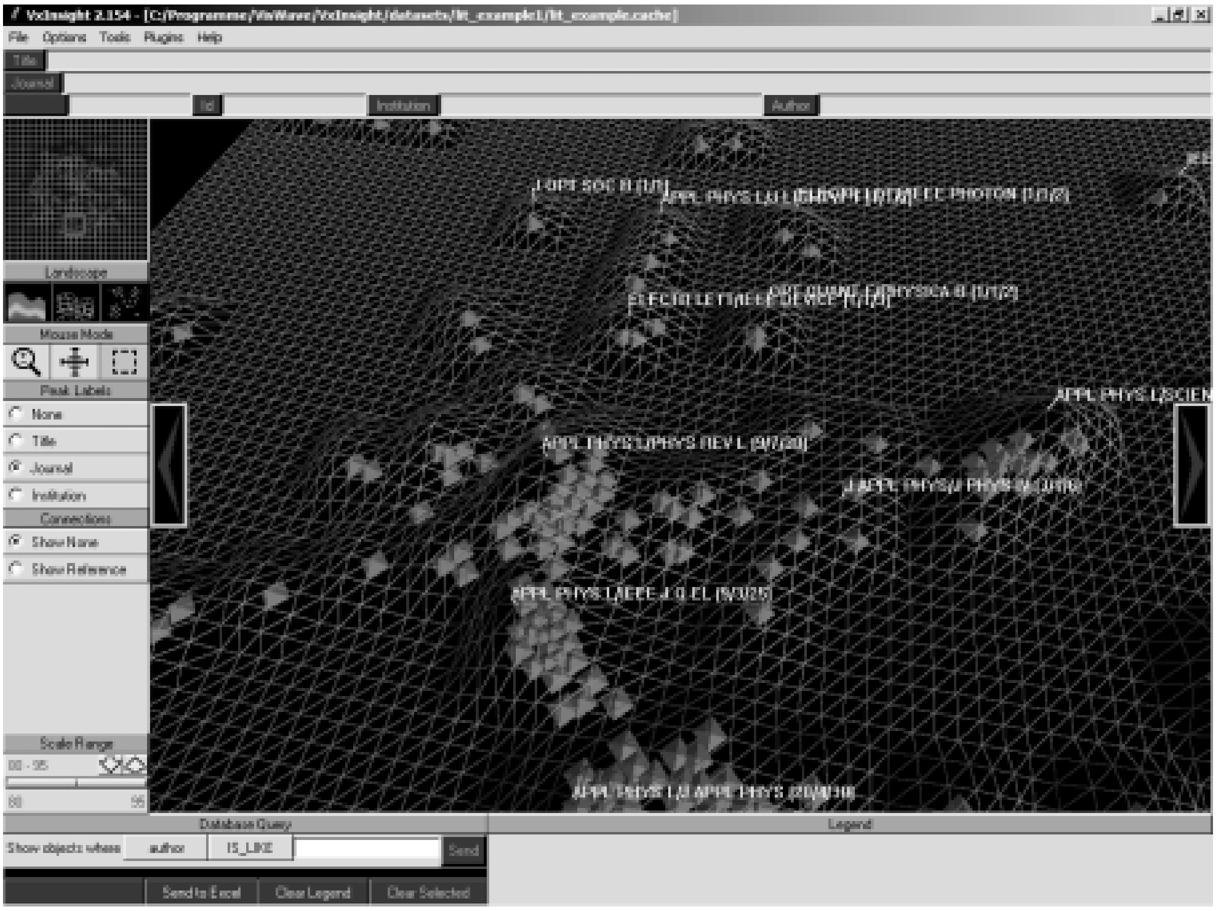

VxInsight 2.154 - [C:/Programme/VxWave/VxInsight/datasets/lit_example1/lit_example.cache]
File Options Tools Plugins Help
Title
Journal
Id
Institution
Author
J OPT SOC B [1/1]
APPL PHYS L/J LIGHTWAVE [1/1/1]
IEEE PHOTON [1/1/2]
ELECTR LETT/IEEE DEVICE [1/1/1]
OPT QUANT E/PHYSICA B [1/1/2]
APPL PHYS L/SCIEN
APPL PHYS L/PHYS REV L [9/7/20]
J APPL PHYS/J PHYS D [3/1/6]
APPL PHYS L/IEEE J Q EL [6/3/25]
APPL PHYS L/J APPL PHYS [20/4/18]
Landscape
Mouse Mode
Peak Labels
○ None
○ Title
● Journal
○ Institution
Connections
● Show None
○ Show Reference
Scale Range
00 - 95
00
95
Database Query
Show objects where
author
IS_LIKE
Send
Legend
Send to Excel
Clear Legend
Clear Selected

Fig. 5. VxInsight displaying a set of articles from physics journals. Similar articles are grouped in clusters. Groups of the same terms in similar documents are used to produce a height field. Hilltops are labelled with the predominant terms (or journal names, in this case).

ing this hierarchical structure can deliver huge improvements in runtime performance and memory usage. Unfortunately, as outlined in Section 2, many document collections are often flat in nature. To overcome this problem, a hierarchy of topics can be extracted automatically by analysing the documents' contents. Once a topical hierarchy has been constructed, a recursive layout algorithm such as InfoSky can be used.

This approach was first implemented and tested as part of a project to cluster and visualise search results on-the-fly called VisIslands [2,16]. More recently, the technique has been applied to an ongoing research project focused on competitive intelligence. To generate the required hierarchical structure from the flat data set, a variant of the well-known k-means clustering algorithm is applied recursively to the data set. The generated hierarchy of clusters is balanced by enforcing limits on the maximum and minimum number of direct children assigned to a cluster, with typical limits being 3 and 12 children, respectively. Since the k-means algorithm has no capability of guessing the number of clusters in advance, strategies for splitting and merging of clusters are implemented along the lines of the ISODATA algorithm [14].

The chosen balancing requirements are derived from usability and performance considerations. From a usability point of view, a cluster with a very large number of children is not well suited for browsing, because users would have to scan through too many objects. With regard to performance, the maximum number of clusters should be limited in order to achieve good performance, since force-directed placement algorithms do not scale well. In this particular case, a very light-weight force-directed placement implementation with cubic runtime behaviour is used: although algorithms with far better scaling are available, the chosen implementation outperforms the alternatives on small datasets. The resulting layout algorithm scales

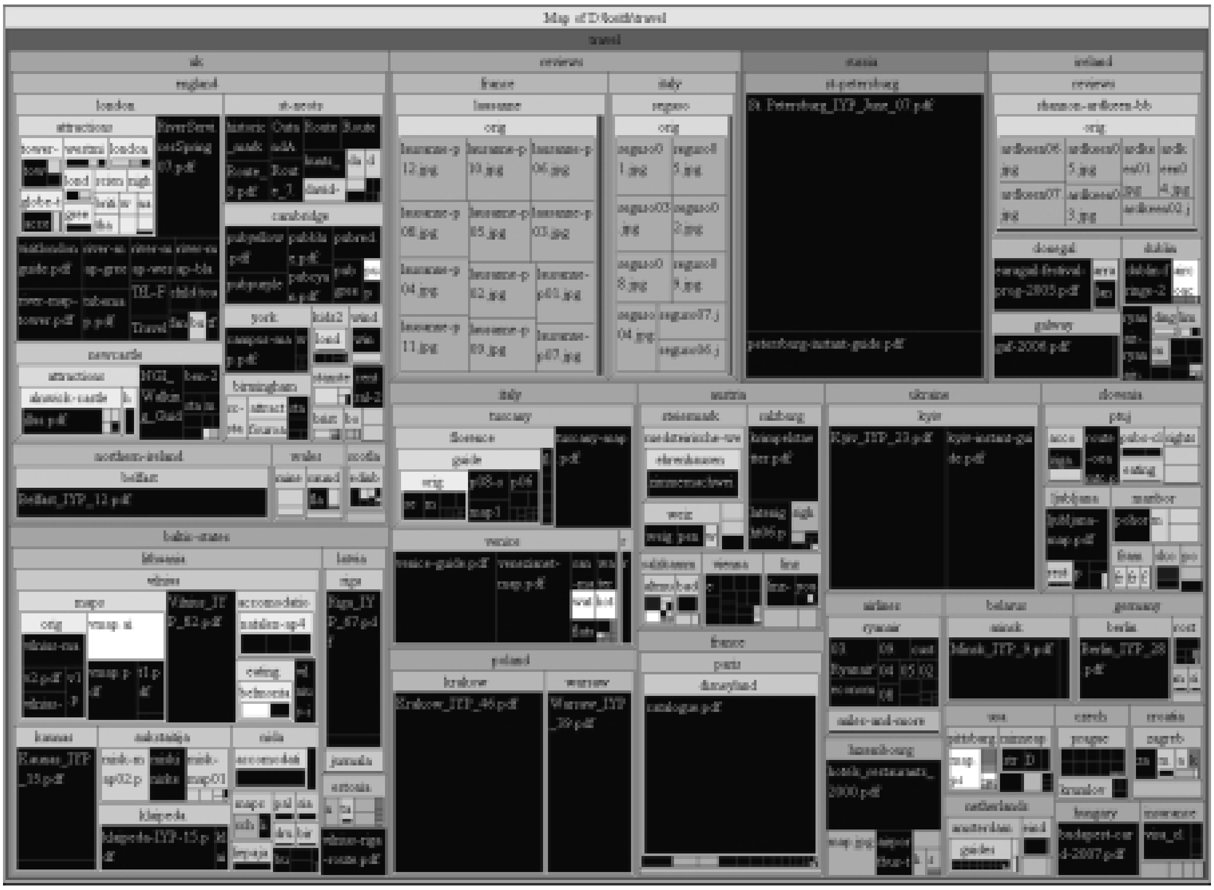


Fig. 6. A TreeMap visualising part of a file system, in this case directories and files collected during travel research. Each solid rectangle represents a file, whose type is indicated by the colour. The enclosing rectangles indicate the directory structure.

very well (approximately $O(n \cdot \log(n))$) and is indeed quite fast. Assuming document term vectors are already available, it takes less than 5 minutes to process 50,000 documents (including the initial hierarchical clustering) on a 3.4 GHz Pentium 4 machine.

A current limitation is that the clustering algorithm requires that all document term vectors are kept in main memory at once, which limits its applicability to approximately 50,000 documents on a machine with 4 gb of main memory. A version employing a clustering algorithm which does not have this requirement, such as BIRCH [25], is currently under development and should be capable of generating similarity layouts of huge data sets on a standard desktop machine.

Figure 9 shows a collection of approximately 30,000 “unstructured” documents visualised as an information landscape generated by the algorithm described above. This representation is similar to the one employed in InfoSky, insofar as similar documents are spatially co-located. Like VisIslands, concepts have been adopted from geographical representations: mountains represent areas where the density of related documents is high, while sparsely populated regions are represented by water. Users can freely navigate within the full 3D virtual landscape or can explore the data set along the hierarchical structure. Keywords extracted from the hierarchical clusters provide labels suited to the chosen level of detail.

Furthermore, the underlying vector space model is capable of handling more than a single vector per document. In addition to the vector space constructed for the full text content, further vector spaces are constructed for extracted entities such as personal names, organisations, or place names. When computing document similarities, the user can assign weights to the different aspects to emphasise relationships of interest. Another interesting possibility is to generate a layout depending on similarities from one vector space and to compute labels from another vector space. This would, for example, allow one to instantly see which organisations

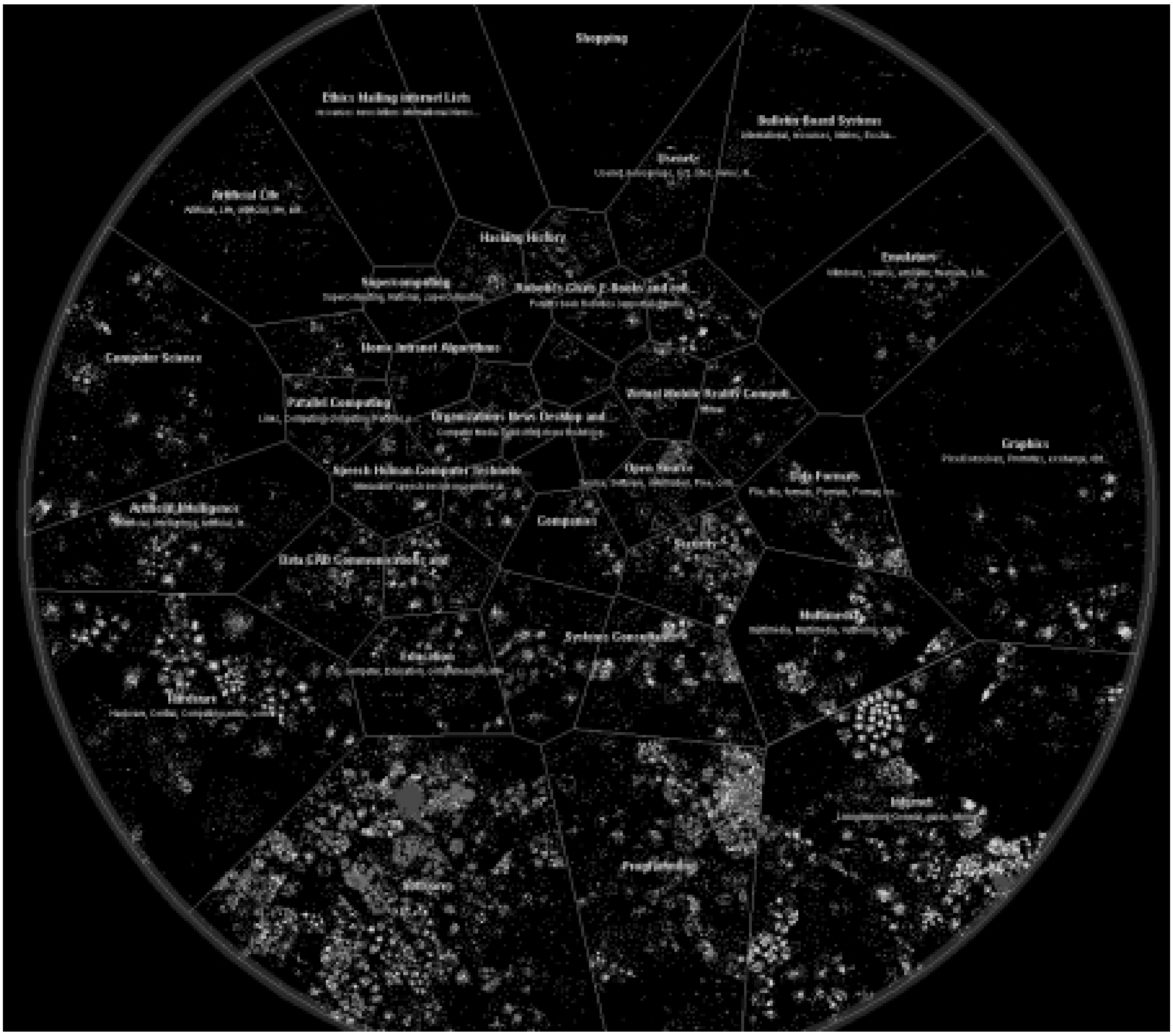


Fig. 7. InfoSky visualising a hierarchy with over 100,000 documents.

produce the most technical papers in a specific field of interest.

4. Visualising time-based collections

In addition to hierarchical structure, temporal information is becoming increasingly important as a dimension along which information can be organised. Discovering trends and causal relationships implicitly present in document collections can be just as important as understanding topical patterns.

The visual representations presented upto now focus on topical relatedness and offer no means of exploring the temporal dimension. ThemeRiver [10] was the first representation for visualising temporal development of topics. Temporal variations of topical clusters, which are encoded by different colours, are shown in the context of external events. The width of a topic changes along the time axis to reveal its “thematic strength” which is typically related to the number of documents referencing that time point.

Another, similar temporal representation, the Timeline Visualisation [12], is shown in Fig. 10. Similar in functionality to ThemeRiver, it differs from ThemeRiver by having constant width and employing polygonal (instead of curved) geometry.

Figure 11 shows time visualisations from the Visual Conversation Analysis (VCA) tool [17]. It is designed for visual analysis of conversations extracted

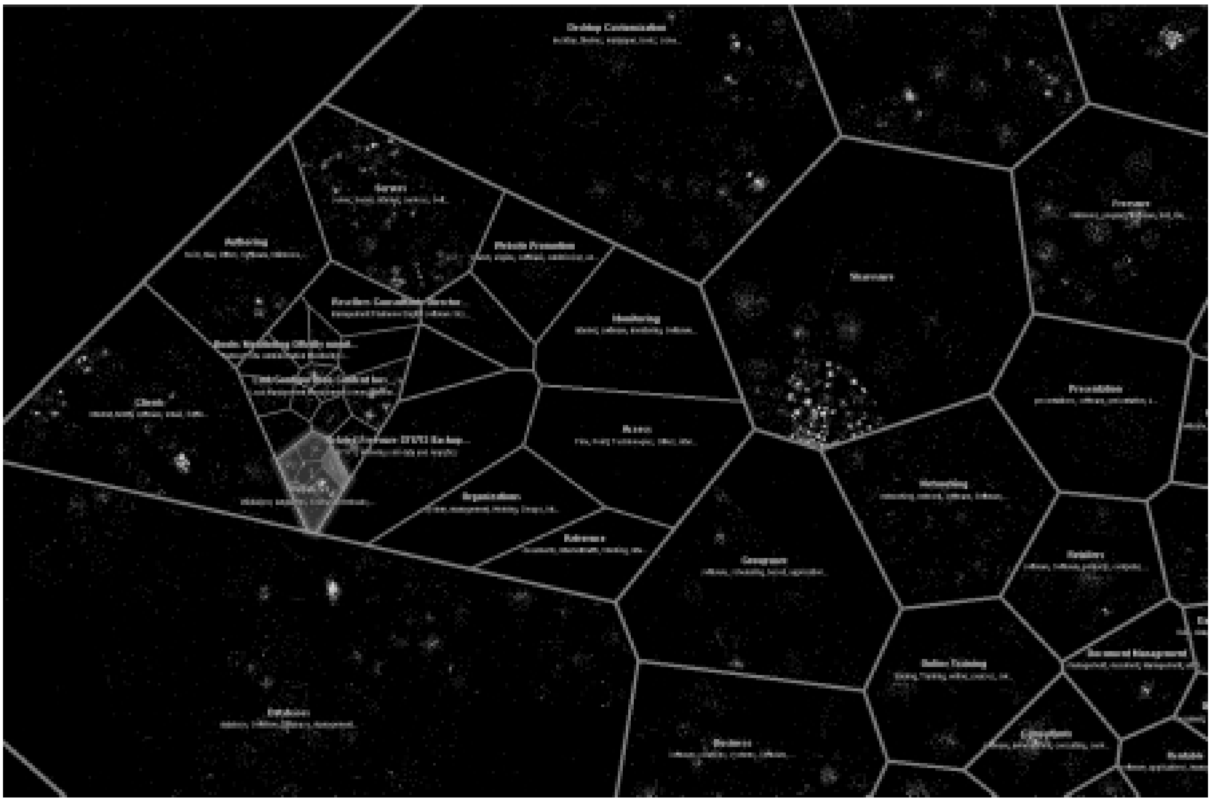


Fig. 8. Zooming in InfoSky. Six levels of zooming into the collection “Webalizer” can be seen in the centre-left.

from video recordings of meetings. All visual components have a shared time axis (the x-axis) which flows from left (past) to right (present). To the top-right there is an interval selection bar which allows the user to navigate in time. Temporal zooming is achieved by changing the width of the interval. To move back and forth in time the user can slide the interval bar, whereby all temporal visualisation components smoothly scroll to the chosen temporal position.

Beneath the interval selection component is a search result bar. Clicking on a result triggers a smooth, animated navigation to the corresponding time point. Towards the bottom of Fig. 11 two thumbnail bars can be seen. The upper thumbnail bar visualises PowerPoint slide transitions, the lower one shows the storyboard. The central components are the activity view and the intensity view. The activity view (upper centre) visualises events produced by people taking part in the meeting over the selected time interval. Rows represent the people (colour-coded), while the rectangles within each row represent events linked to the corresponding person, in this case speech activity of the participants. The width of each rectangle represents the duration of the corresponding event, a zero duration event being represented by a vertical line. The view provides a clear representation of event boundaries, but is not appropriate for visualising overlapping events.

The intensity view (lower centre) visualises the intensity of overlapping events over the selected time interval. Here, each row corresponds to a different thematic topic. Overlapping events are stacked over each other and added together to yield higher "hills", so that the height of a hill indicates a higher density of events. In contrast to the activity view, the intensity view is not suitable for exact representation of event boundaries.

The representations described so far in this section focus on the discovery of trends and relationships within the temporal dimension. In contrast to visualisations based on static similarity layouts, they cannot convey topical relationships. The question arises as to whether it would be possible to combine visual components representing topical relationships, such as information landscapes, with components designed to represent temporal information to allow for simultaneous topical-temporal analysis.

One way to address this problem would be to tightly couple temporal views with a dynamic topography

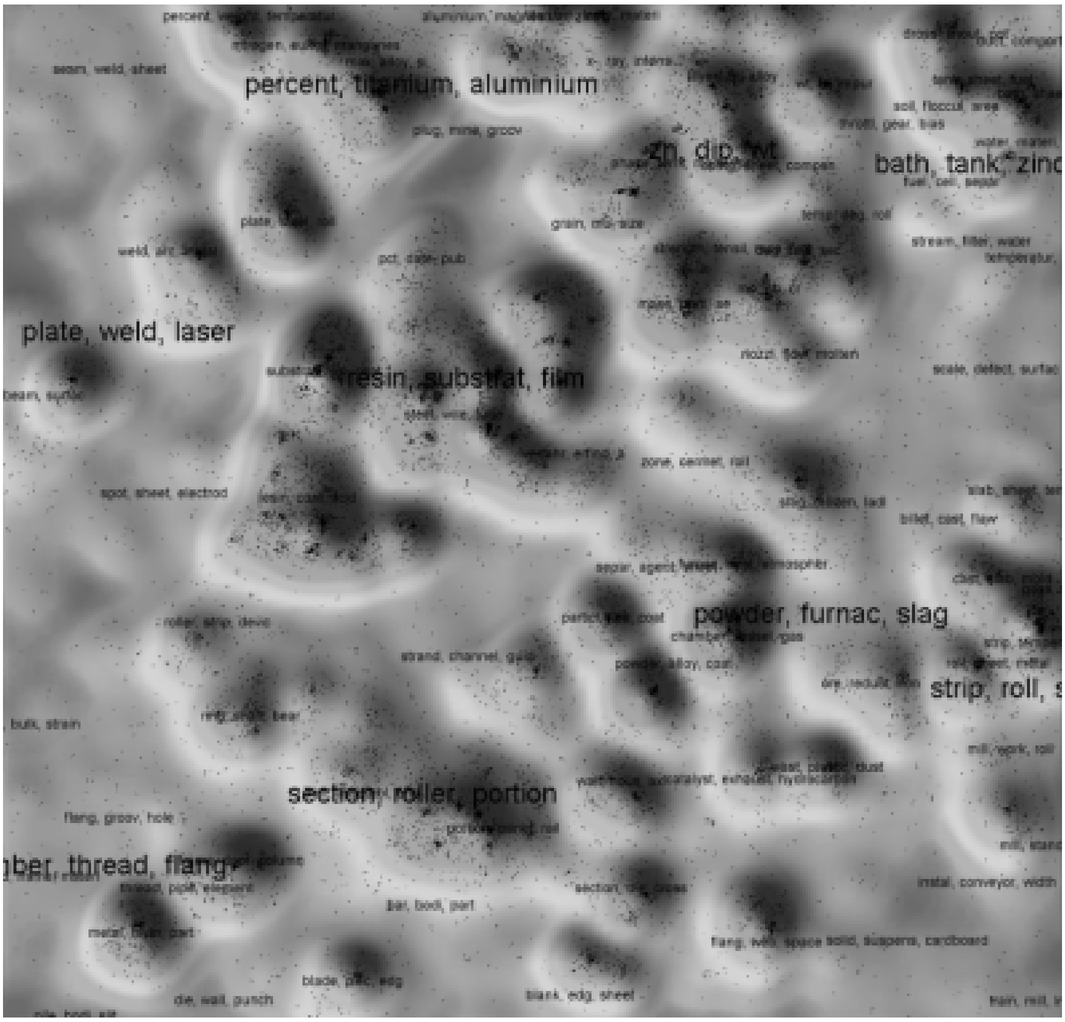


Fig. 9. An information landscape showing a flat collection of approximately 30,000 documents.

information landscape. To understand the idea, one might imagine the surface of the landscape placed orthogonally to the temporal axis, as shown in Fig. 12, so that the surface of the landscape becomes a slice of the temporal view(s) for the selected time interval. As the user modifies the selected time interval, the topography of the landscape is dynamically updated to visualise only the selected subset of the data. As documents are removed or added to the current active set, old islands and hills may disappear or change their shape and new ones might arise from the sea. Hills may drift towards one other (corresponding to the merging of previously separate topical clusters) or an island may split off (corresponding to the break-up of a cluster).

In such a scenario, topography transitions must be performed incrementally, in the sense that only those changes should be introduced which are really necessary. Regions of the visualisation which are little affected by the choice of the time interval should remain as stable as possible, To support the user in understanding modifications to the topography, incremental transitions should be smoothly animated so that the user can follow and understand the changes. Such dynamic, incremental, animated information landscapes were introduced in [18], but were only applied to very small data sets up to a few hundred documents. Applying the concept to large data sets poses substantial challenges and is the subject of ongoing work [22].

5. Concluding remarks

The techniques presented in this paper support the visual exploration of large text collections. For text

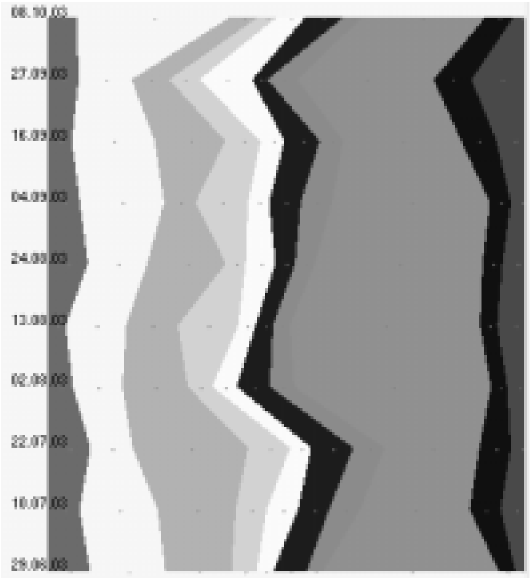


Fig. 10. Temporal thematic visualisation from the TimelineVis system.

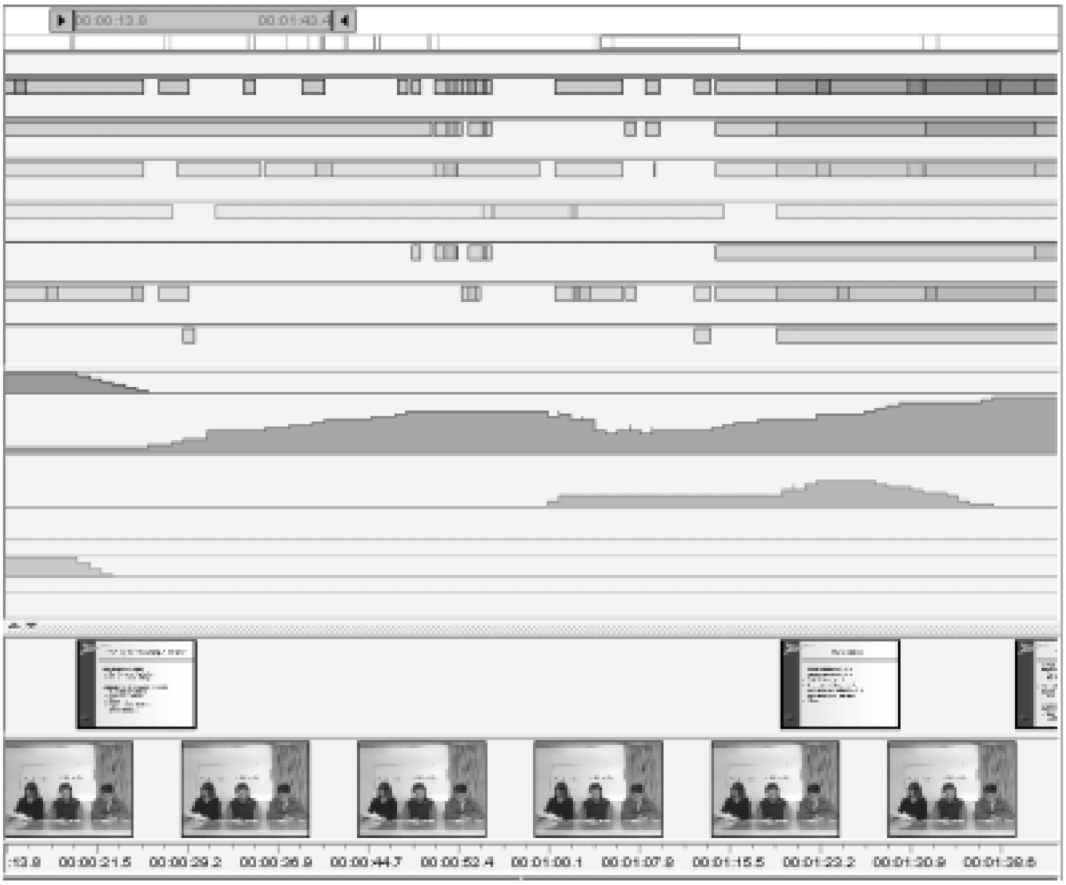


Fig. 11. Temporal visualisation of speech in the Visual Conversation Analysis tool.

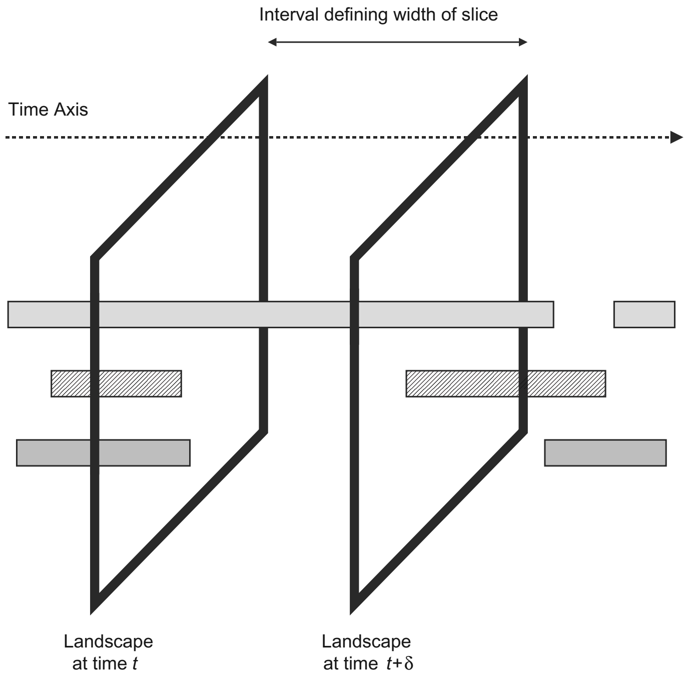


Fig. 12. Idea behind generation of information landscapes with dynamic topography.

documents, a similarity matrix is constructed to express the (dis)similarity between every pair of documents. The similarity matrix is then projected to a 2d or 3d display space.

The techniques presented here generalise to other kinds of collection in addition to text. For more general object collections, feature vectors based on other features extracted from the objects are used to construct a vector space. For example, for collections of images, features such as the frequencies of particular colours, or texture-based features such as coarseness, contrast, and directionality can be used [7]. For collections of 3d models, features such as the bounding box, the distribution of normal vectors, and ray-based moments can be used [4].

Acknowledgments

The Know-Center is funded within the Austrian COMET Program – Competence Centers for Excellent Technologies – under the auspices of the Austrian Ministry of Transport, Innovation and Technology, the Austrian Ministry of Economics and Labor and by the State of Styria. COMET is managed by the Austrian Research Promotion Agency FFG.

References

[1] K. Andrews, *Information Visualisation: Lecture Notes*, 2007. http://courses.iicm.tugraz.at/ivis/ivis.pdf.

[2] K. Andrews, C. G.ütl, J. Moser, V. Sabol and W. Lackner, Search Result Visualisation with xFIND. In *Proc. International Workshop on User Interfaces to Data Intensive Systems* (*UIDIS 2001*), pp. 50–58. IEEE Computer Society Press, Zurich, Switzerland, May 2001. doi:10.1109/UIDIS.2001.929925.

[3] K. Andrews, W. Kienreich, V. Sabol, J. Becker, G. Droschl, F. Kappe, M. Granitzer, P. Auer and K. Tochtermann, The InfoSky Visual Explorer: Exploiting Hierarchical Structure and Document Similarities, *Information Visualization* **1**(3/4) (December 2002), 166–181. doi:10.1057/palgrave.ivs.9500023.

[4] B. Bustos, D.A. Keim, D. Saupe, T. Schreck and D.V. Vranic, *Feature-based Similarity Search in 3D Object Databases. ACM Computing Surveys*, December 2005. ISSN 0360-0300. doi:10.1145/1118890.1118893. http://infovis.uni-konstanz.de/research/projects/SimSearch3D/publications/csur05.pdf.

[5] M. Chalmers and P. Chitson, Bead: Explorations in Information Visualization. In *Proc. SIGIR '92*, pp. 330–337. ACM, Copenhagen, Denmark, September 1992. doi:10.1145/133160.133215. http://www.dcs.gla.ac.uk/?matthew/papers/sigir92.pdf.

[6] G.S. Davidson, B. Hendrickson, D.K. Johnson, C.E. Meyers, and B.N. Wylie, Knowledge Mining with VxInsight: Discovery Through Interaction, *Journal of Intelligent Information Systems* **11**(3) (November 1998), 259–285. doi:10.1023/A:1008690008856. http://www.cs.sandia.gov/projects/VxInsight/pubs/jiis98_prepub.pdf.

[7] T. Deselaers, D. Keysers and H. Ney, *Features for Image Retrieval: An Experimental Comparison. Information Retrieval,*

December 2007. ISSN 1573-7659. doi:10.1007/s10791-007-9039-3. http://citeseer.ist.psu.edu/deselaers04features.html.

[8] P. Eades, A Heuristic for Graph Drawing, *Congressus Numerantium* **42** (1984), 149–160. http://www.cs.usyd.edu.au/˜peter/old_spring_paper.pdf.

[9] T.M.J. Fruchtermann and E.M. Reingold, Graph Drawing by Force-Directed Placement, *Software: Practice and Experience* **21**(11) (November 1991), 1129–1164. doi:10.1002/spe.4380211102.

[10] S. Havre, E. Hetzler, P. Whitney and L. Nowell, ThemeRiver: Visualizing Thematic Changes in Large Document Collections, *IEEE Transactions on Visualization and Computer Graphics* **8**(1) (January 2002), 9–20. doi:10.1109/2945.981848.

[11] F. Jourdan and G. Melancon, Multiscale Hybrid MDS. In *Proc. Eighth International Conference on Information Visualisation* (*IV'04*), pp. 388–393. IEEE, London, UK, July 2004. doi:10.1109/IV.2004.1320173.

[12] W. Kienreich, V. Sabol, M. Granitzer, W. Klieber, M. Lux and W. Sarka, A Visual Query Interface for a Very Large Newspaper Article Repository. In *Proceedings of the Sixteenth International Workshop on Database and Expert Systems Applications* (*DEXA 2005*), pp. 415–419. IEEE Computer Society Press, Copenhagen, Denmark, August 2005. ISBN 0769524249. doi:10.1109/DEXA.2005.35.

[13] M. Krishnan, S.J. Bohn, W.E. Cowley, V.L. Crow and J. Nieplocha. Scalable Visual Analytics of Massive Textual Datasets. In *Proc. Parallel and Distributed Processing Symposium* (*IPDPS 2007*), pp. 1–10. IEEE, Long Beach, California, USA, March 2007. doi:10.1109/IPDPS.2007.370232. http://infoviz.pnl.gov/pdf/inspire_ipdps.pdf.

[14] N. Memarsadeghi, D.M. Mount, N.S. Netanyahu and J. Le Moigne, A Fast Implementation of the Isodata Clustering Algorithm, *International Journal of Computational Geometry & Applications* **17**(1) (February 2007), 71–103. doi:10.1142/S0218195907002252. http://www.cs.umd.edu/˜mount/Projects/ISODATA/ijcga06-isodata.pdf.

[15] F.V. Paulovich, M.C.F. Oliveira and R. Minghim, The Projection Explorer: A flexible tool for Projection-Based Multidimensional Visualization. In *Proc. XX Brazilian Symposium on Computer Graphics and Image Processing* (*SIBIGRAPI 2007*), pp. 27–36. IEEE, Belo Horizonte, Brazil, October 2007. doi:10.1109/SIBGRAPI.2007.39. http://infoserver.lcad.icmc.usp.br/infovis2/PEx.

[16] V. Sabol, *Visualisation Islands: Interactive Visualisation and Clustering of Search Result Sets*, Master's thesis, Graz University of Technology, Austria, October 2001. http://ftp.iicm.edu/pub/theses/vsabol.pdf.

[17] V. Sabol, C. Gütl, T. Neidhart, A. Juffinger, W. Klieber and M. Granitzer, Visualization Metaphors for Multi- Modal Meeting Data. In *Proc. 6th Workshop of the Multimedia Metadata Community* (*WMSRM07*), pp. 250–269. Verlagshaus Mainz, Aachen, Aachen, Germany, March 2007. ISBN 3861309297.

[18] V. Sabol, W. Kienreich, M. Granitzer, J. Becker, K. Tochtermann and K. Andrews, Applications of a Lightweight, Web- Based, Retrieval, Clustering and Visualisation Framework. In *Proc. Practical Apsects of Knowledge Management* (*PAKM 2002*), pp. 359–368. Springer LNCS 2569, Vienna, Austria, December 2002. http://www.springerlink.com/content/pjuxn5qjuyf9eta1/.

[19] G. Salton, *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*, Addison-Wesley, August 1988. ISBN 0201122278.

[20] B. Shneiderman, Tree Visualization with Tree- Maps: A 2-d Space-Filling Approach, *ACM Transactions on Graphics* **11**(1) (January 1992), 92–99. doi:10.1145/102377.115768.

[21] R. Spence, *Information Visualization: Design for Interaction*, Prentice Hall, Second edition, December 2006. ISBN 0132065509.

[22] W.K.V. Sabol and M. Granitzer, Fused Exploration of Temporal Developments and Topical Relationships in Heterogeneous Data Sets. In *Proc. 11th International Conference on Information Visualisation* (*IV'07*), pp. 369–375. IEEE Computer Society Press, Zurich, Switzerland, July 2007. ISBN 0769529003. doi:10.1109/IV.2007.53.

[23] C. Ware, *Information Visualization: Perception for Design*, Morgan Kaufmann, 2nd edition, 2004. ISBN 1558608192.

[24] J.A. Wise, The Ecological Approach to Text Visualization, *Journal of the American Society for Information Science* **50**(13) (November 1999), 1224–1233. doi:10.1002/(SICI)1097-4571(1999)50:13!1224::AID-ASI8?3.0.CO;2-4. http://www.interscience.wiley.com/cgi-bin/abstract/66001476/ABSTRACT.

[25] T. Zhang, R. Ramakrishnan and M. Livny, BIRCH: An Efficient Data Clustering Method for Very Large Databases. In *Proc. 1996 ACM SIGMOD International Conference on Management of Data* (*SIGMOD '96*), pp. 103–114. ACM, Montreal, Quebec, Canada, June 1996. ISBN 0897917944. doi:10.1145/233269.233324. http://citeseer.ist.psu.edu/zhang96birch.html.