Dynamic Topography Information Landscapes – An Incremental Approach to Visual Knowledge Discovery

Kamran Ali Ahmad Syed[1], Mark Kröll[2], Vedran Sabol[2],
Arno Scharl[3], Stefan Gindl[3], Michael Granitzer[4], and Albert Weichselbraun[5]

[1] Vienna University of Economics and Business, Institute for Information Business, Vienna, AT
kamran.syed@wu.ac.at
[2] Know-Center, Division for Knowledge Relationship Discovery, Graz, AT
{mkroell,vsabol}@know-center.at
[3] MODUL University Vienna, Department of New Media Technology, Vienna, AT
{arno.scharl,stefan.gindl}@modul.ac.at
[4] University of Passau, Media Informatics Department, Passau, DE
michael.granitzer@uni-passau.de
[5] University of Applied Sciences Chur, Faculty of Information Science, Chur, CH
albert.weichselbraun@htwchur.ch

Abstract. Incrementally computed information landscapes are an effective means to visualize longitudinal changes in large document repositories. Resembling tectonic processes in the natural world, dynamic rendering reflects both long-term trends and short-term fluctuations in such repositories. To visualize the rise and decay of topics, the mapping algorithm elevates and lowers related sets of concentric contour lines. Addressing the growing number of documents to be processed by state-of-the-art knowledge discovery applications, we introduce an incremental, scalable approach for generating such landscapes. The processing pipeline includes a number of sequential tasks, from crawling, filtering and pre-processing Web content to projecting, labeling and rendering the aggregated information. Incremental processing steps are localized in the projection stage consisting of document clustering, cluster force-directed placement and fast document positioning. We evaluate the proposed framework by contrasting layout qualities of incremental versus non-incremental versions. Documents for the experiments stem from the blog sample of the *Media Watch on Climate Change* (www.ecoresearch.net/climate). Experimental results indicate that our incremental computation approach is capable of accurately generating dynamic information landscapes.

Keywords: Information visualization, information landscape, incremental clustering, multi-dimensional scaling.

1 Introduction

These days we are confronted not only with constantly growing, but also with continuously and often rapidly changing "big data" repositories. Information Landscapes represent a powerful visualization technique for conveying topical relatedness in large

A. Cuzzocrea and U. Dayal (Eds.): DaWaK 2012, LNCS 7448, pp. 352–363, 2012.

document repositories [20]. Yet, the concept of information landscapes does only allow for visualizing static conditions. In previous research, we have introduced dynamic topography information landscapes [29] to address both (i) topical relatedness and (ii) visualization of data changes. As such, dynamic landscapes have proved valuable in enterprise scenarios involving visual knowledge discovery in large, dynamic text repositories, where they have been applied for tracking of topical relationships and trends in media and patent databases [30].

Dynamic topography information landscapes are visual representations based on a geographic map metaphor where topical relatedness is conveyed through spatial proximity in the visualization space with hills representing agglomerations (clusters) of topically similar documents. Hills are labeled with dominant terms from the underlying documents to facilitate the users' orientation. When a document repository changes over time, e.g. new documents are added or old documents are removed, the overall topical structure changes as well. Dynamic information landscapes convey these changes as tectonic processes which modify the landscape topography accordingly. In the process of generating information landscapes, high-dimensional data is projected into a lower-dimensional space. Yet, existing dimensionality reduction approaches lack several aspects including (i) support for incremental computation, (ii) scalability with respect to data set size and high-dimensionality (iii) and generation of aesthetically pleasing layouts which are necessary for visual applications.

This paper presents an incremental, scalable algorithmic approach for computing dynamic topography information landscapes capable of visualizing dynamically changing text repositories. Our incremental processing pipeline is introduced in Section 3 and includes implementation details of text preprocessing, projection (dimensionality reduction), labeling and rendering stages where the projection part combines document clustering, cluster force-directed placement and, an improved approach to fast document positioning. We conclude this section by visualizing a temporal sequence of eight incrementally computed information landscapes, which reflect weekly changes in the underlying document set. In Section 4, we experimentally verify our approach's runtime behavior which we discuss only in theory in Section 3. In addition, we evaluate our incremental computation framework by comparing stress values between incrementally and non-incrementally computed layouts. Documents for these experiments are taken from the environmental blog sample of the *Media Watch on Climate Change* [16], a Web content aggregator on climate change and related environmental issues. Our experimental results show that the incremental computation approach yields not only comparable, but even slightly better stress values and thereby indicate our framework's validity.

2 Related Work

Information landscapes are commonly used to visualize topical relatedness in large document repositories, for example in Krishnan et al. [20] and Andrews et al. [1]. Static landscape visualizations, however, cannot convey changes. *ThemeRiver* [13] is a visual representation designed to represent changes in topical clusters, but it cannot

express relatedness between documents or topical clusters. Visualization of topical changes through information landscapes with dynamic topographies were proposed in Sabol et al. [28]. An approach suitable for larger data sets was demonstrated in [27]. It relies on 3D acceleration for animated morphing of landscape geometry, which makes it unsuitable for Web applications. However, the performance of the incremental algorithms remains unclear as it was not evaluated or compared with a non-incremental variant.

Visualization techniques in general have to cope with today's ever-growing data production and data consumption. Incremental algorithms provide the required functionality to process big data. Incremental algorithms do not recalculate their internal model from scratch for newly arriving data items and are thus capable of efficiently handling and seamlessly integrating continuously changing or growing data. In the context of generating dynamic information landscapes we review work on incremental dimensionality reduction and incremental clustering techniques.

Incremental Dimensionality Reduction. Dimensionality reduction techniques transform high-dimensional data into low-dimensional data seeking to lose as little information as possible. This transformation has turned out to be particularly useful in the field of visualization for projecting the high-dimension data into the low-dimensional visualization space. To face the growing amount of data, incremental variants have been developed usually on top of batch methods. Incremental unsupervised techniques include multi-dimensional scaling (cf. [4]), singular value decomposition (cf. [31]), principal component analysis (cf. [2]), random indexing (cf. [18]) or locally linear embeddings (cf. [19]). Unsupervised methods are effective in finding compact representations, but ignore valuable class label information of the training data. Incremental supervised techniques are thus better suited for pattern classification tasks. Representatives of incremental supervised dimensionality reduction techniques include linear discriminant analysis (cf. [23]) or subspace learning (cf. [34]).

Incremental Clustering. Incremental clustering algorithms can be traced back to the 1970s, cf. Hartigan's *leader* algorithm which requires only one pass through the data [12], Slagle's shortest spanning path algorithm [33] or Fisher's COBWEB system, an incremental conceptual clustering algorithm [9]. The COBWEB system, for example, has been successfully applied to support fault diagnosis or bridge design. Inspired by COBWEB, Gennari et al. proposed the CLASSIT [11] system which is capable of handling numerical data sets. In [5], the authors introduced an incremental clustering algorithm for dynamic information processing. In dynamic databases there is a constant adding or removing of data items over time. The idea is that these changes should be recognized in the generated partition without affecting current clusters. In the late nineties, several incremental clustering algorithms have been presented including BIRCH [35], incremental DBSCAN [8] to support data warehousing or Ribert et al.'s clustering algorithm to generate a hierarchy of clusters [26]. Incremental clustering of text documents has been conducted as a part of the Topic Detection and Tracking initiative [1] to detect a new event from a stream of news articles.

To compute dynamic topography information landscapes in an incremental and thus timely efficient manner, we integrate and combine incremental aspects into the

generation process. (i) For clustering, we apply a simple, spherical k-means [7] and use previously computed partitions of the document set as initial state for incremental computations. (ii) We introduce an improved approach for document positioning which is essentially based on a simple spring forces-based model (cf. [10]) since we observed that landscapes generated with standard positioning method displayed geometrical edges. (iii) We use a force-directed placement (FDP) algorithm [10] for projecting these high-dimensional cluster centroids into a 2D visualization space. The parameters of FDP-based methods provide significant control over the layout, which allows them to deliver more pleasing layouts than traditional methods. The FDP algorithm is intrinsically incremental when applied on a previously computed stable layout. Re-applying FDP on a previous layout of centroids with modified similarities will produce a new layout closely resembling the previous one.

3 Algorithmic Approach

In this section we introduce and describe our approach to generating dynamic information landscapes. Fig. 1 depicts the overall workflow, which can be grouped into three main components: (i) First (shown in green), we prepare an augmented document-term matrix by combining information from keyword relevance and word frequency tables. (ii) In a second step (cyan), we cluster and position the documents. We use the k-means clustering algorithm to partition the documents into topically related clusters. We then employ force directed placement to project clusters centroid positions into 2D visualization space, and apply a fast method for positioning documents in 2D based on cluster positions. (iii) In the last step (magenta), we use the documents' layout position to model a topical landscape which is essentially an elevation matrix on a 2D grid. A coloring scheme is used to construct landscape surface images. A peak detection algorithm then finds major peaks (hills) and collects underlying documents to compute text descriptors for labeling the peaks. Note that previous computation results are used as initial state for incremental processing (in orange).

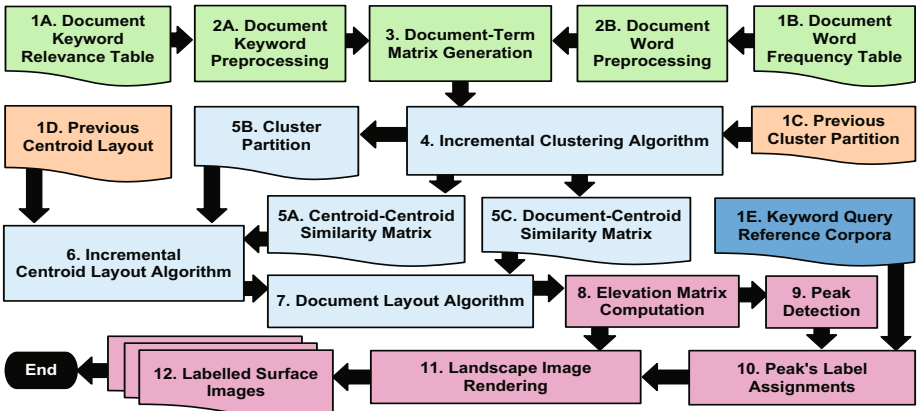

Fig. 1. Workflow diagram for the incremental landscape computation framework

Details on these three main components will be provided in the following subsections 3.1 to 3.3, followed by a separate description of the architecture's incremental aspects in Section 3.4.

3.1 Document-Term Matrix Generation

Prior to the beginning of the computation, the raw textual data to be analyzed is gathered via a web crawler, then converted and annotated into the content repositories based on previous research [16], [32]. We utilize our experiences with *webLyzard*, an established and scalable media monitoring and Web intelligence platform (www.weblyzard.com), to generate the document keyword relevance table (1A) and the document word frequency table (1B).

Using the information from 1A and 1B, we create the document keyword matrix (2A) as well as the document word matrix (2B). Both matrices are then linearly combined into one augmented document term matrix (3) with unique terms IDs.

3.2 Clustering and Projection

The incremental clustering algorithm (4) takes the document term matrix (3) as input and outputs (i) a centroid-to-centroid similarity matrix (5A), (ii) a document-centroid relationship graph (5B), and (iii) a documents-to-centroids similarity matrix (5C).

In incremental mode the k-means algorithm module is initialized by the previously computed clustering result (1C). The centroid positioning algorithm (6) uses results (5A) and (5B). The algorithm can be initialized with previous centroid positions (1D) for the incremental case, or by assigning random positions for the non-incremental computation. The centroid positions (6) and the document to centroid similarity matrix (5C) are then used for computing the document positions (7).

Document Clustering. We apply the spherical k-means algorithm [7] to partition the documents into topical clusters. The k-means algorithm is known to be highly sensitive to the initial guess of the cluster partitions and the number of partitions. To overcome this sensitivity, we use the k-means++ seeding method [3]. In addition, we split and merge the clusters [22] for deducting the number of clusters within the limit of specified minimum and maximum bounds. As human cognition puts certain limits to conceiving visualizations, we limited the number of clusters to account for usability. We observe that setting the minimum and maximum number of cluster bounds to be 30 and 40, respectively, result in meaningful and aesthetically pleasing information landscapes. Therefore, in subsequent iterations we perform the splitting of large clusters to obtain higher cluster cohesion as well as the merging of small, similar clusters, according to improvements using Bayesian Information Criterion [24].

The algorithm's runtime complexity is $O(mnd)$, where m is the number of clusters, n is the number of documents, and d is the dimensionality of the term space. Since $m \ll n$ in our case, and according to Heaps' law [14], d scales logarithmically with n, the clustering part of our algorithm is considered to scale with $O(n \log(n))$. For incremental clustering, we use previously computed partitions of the document set

as the initial state. For a fixed number of documents to be clustered incrementally, old documents are removed from their respective clusters and new documents are added to the most similar cluster centroid. Afterwards, additional k-means iterations, including the split and merge procedure, are performed to further refine the initial partition.

Cluster Positioning. The partitioned set of documents is represented by the high-dimensional centroids of the respective clusters. We use a force-directed placement (FDP) algorithm [10] for projecting these high-dimensional cluster centroids into a 2D visualization space. The idea is that attractive forces pull together topically similar centroids while dissimilar centroids are repulsed. Spatial closeness between centroids thus relates to their topical closeness. The FDP algorithm is known to produce accurate and aesthetically pleasing layouts. Most FDP variants scale poorly, e.g. $O(m^3)$. Yet, as in our approach $m \ll n$, and because there is a fixed upper limit on m, in our case 40 clusters, the runtime complexity of cluster positioning may be considered constant. As stopping criterion for the FDP algorithm, we used two parameters: (i) a fixed maximum number of iterations, and (ii) the local minima for the stress value [21]. The most attractive feature of the FDP algorithm is that it is intrinsically incremental when applied on a previously computed stable layout. The impact of incremental clustering is reflected in similarities between cluster centroids. When changes in the data set are small, the similarities between the centroids will also change by a small proportion. Re-applying FDP on previous layout of centroids with modified similarities will produce a new layout closely resembling the previous one.

Document Positioning. In an earlier version of the algorithm [29], we used an algorithm based on Delaunay triangulation of centroid positions in the 2D space. The most similar triangle was chosen based on the similarities between the document and the most similar centroids, and the document position was assigned using Barycentric coordinates in $O(m)$ time, m being the number of centroid vertices.

Unfortunately, we observed that landscapes generated with this positioning method reflected geometrical edges; i.e., documents were positioned in straight lines. To maintain the viewing experience of a realistic landscape without artifacts, and to achieve a linear running time, we introduce an improved approach for fast document positioning which is essentially based on a simple spring forces-based model (cf. [10]). In this model we assumed that the document, in two dimensions, is attached to each centroid, in two dimensions, by a spring having a spring constant proportional to the similarity between the document and the centroid in the n-dimensional space.

If $R_1, R_2, R_3, \ldots, R_m$ are the given position vectors of all m centroids and $s_{i1}, s_{i2}, s_{i3}, \ldots, s_{im}$ are the given similarities of the i^{th} document with m centroids respectively, then an analytical solution of the equilibrium conditions for Hooke's law forces between i^{th} document and all centroids eventually formulate the position of the i^{th} document as $r_i = \sum_{k=1..m} s_{ik} R_k / \sum_{k=1..m} s_{ik}$. This simple algorithm makes our computation for document positioning linear in time and, in contrast to [29], without any overhead

3.3 Landscape Creation and Peak Labeling

With document layout positions at hand, we compute an elevation matrix (8) that represents an information landscape model. We then utilize this matrix to identify

peak locations, heights and a list of documents related to the peak (9). The peak detection employs a kernel window convolution over the landscape model. The peak label assignment module (10) determines the peak's labels by using the list of documents under the peak for querying and comparing with the semantically tagged reference corpora (1E), which is continuously refined by the *webLyzard* platform. Finally, the assigned labels are positioned on the information landscape surface images (12), computed based on the coloring scheme (8) and the heuristic labeling algorithms of the landscape image rendering module (11).

Landscape Modeling. Information landscapes with specific resolutions are modeled as elevation matrices of the same resolution. A document is thought of as a small Gaussian peak at the corresponding position on the underlying matrix cells. The influence of a document on a matrix cell location is reflected by the value of Gaussian density at that location. Thus the height and the asymptotic radius of the Gaussian peak reflect the document's influence in the landscape. We further assume a document has a fixed influence on its own location on the matrix cell. The densities of all documents at particular location are superimposed, adding to the elevation values of the underlying matrix cells.

Peak Detection. A kernel window-based peak detection algorithm is used to detect the significant peaks of the landscape (cf. [15], [25]). The average of the convolution of the window with the elevation matrix is compared with the center value of the matrix cell. A peak is assumed if the center value is higher than the average convolving value. After detecting the significant peaks, documents are assigned to their nearest peak by using the minimum Euclidean distance criterion in the 2D layout.

Label Computation. The term distribution in the set of documents in the vicinity of a peak is compared with a reference distribution. A chi-square test of significance with Yates' correction determines over-represented terms. The *term* co-occurrence analysis, based on pattern matching algorithm, along with *trigger phrases* based on regular expressions, is used to identify the frequently appearing text fragments within the same sentences and within the documents [16], [32]. The redundancy of nouns' singular and plural forms and synonyms in the resultant list of labels are removed by using a combination of regular expression queries and WordNet library lookup.

Map Generation and Label Placement. In the final step, colors are assigned to the image pixels depending on the density of the corresponding density matrix cells. In our scheme of colors the blue is used to express lowest density, then green and brown, and finally light gray is used for highest density. The resulting landscape surface image resembles a geographic map with peaks at areas, where document density is large, and oceans or valleys, where document density is low. Finally, a heuristic point feature label placement algorithm [6] is used with the labeling quality evaluation in the following basic rules: (i) No overlap of a label with other labels and the image boundary. (ii) No overlap of a label and another peak location. (iii) Each label is placed among the four possible labeling rectangular spaces of the peak locations. (iv) At most five labels for a peak location can be assigned.

3.4 Incremental Computation

Computing incremental landscapes at first requires an initial computation of a landscape. We apply our algorithm to an initial data set where the documents' layout positions are saved for future use. Whenever the data set changes, the incremental k-means algorithm is initialized with a previously computed stable partition, i.e. with the old locations for the new centroids. The ongoing process of removing old documents and adding new documents to the most similar clusters leads to several k-means iterations for the next stable partition. The successive iteration of FDP will stop at the first local minima for the average stress value.

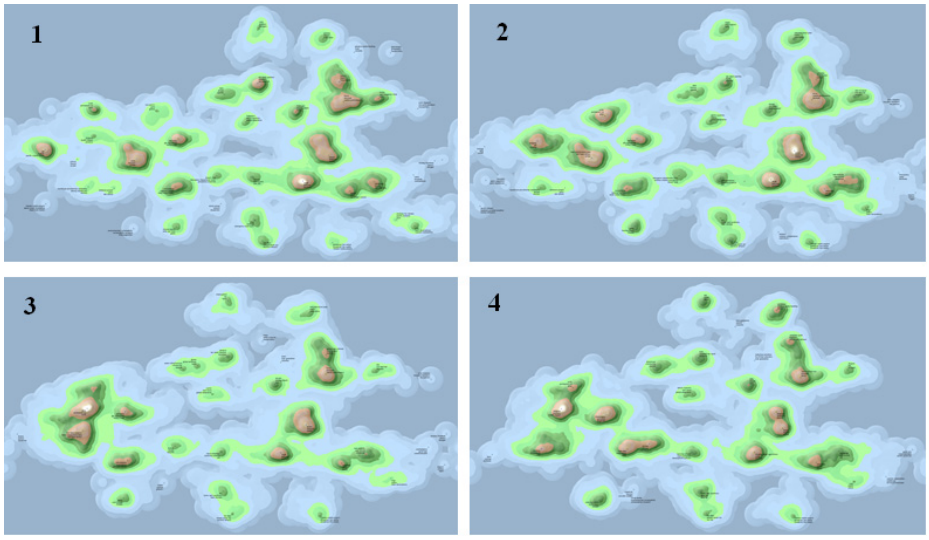

Fig. 2. A sequence of incrementally computed landscapes from environmental blogs, visualizing 2,000 documents each, reflects weekly changes from Sept 30th, 2011 to Oct 21th 2011. Approximately 10% of the data set changed between each individual step resulting in seamless transformations of topography portions, while the overall structure remains stable. Rising hills indicate the emergence of new topics (images 1, 2 and 3); shrinking hills a fading of topics (image 4). Hill movements towards or apart from each other indicates converging or diverging topical clusters. As the incremental algorithm seamlessly integrates a stream of continuous changes, the user retains orientation through recognition of unchanged parts of the topography.

To acknowledge the growing number of documents to be processed by state-of-the-art web intelligence applications, we briefly discuss scalability issues in this section. Many processing steps of our algorithmic approach scale linearly (or even better) with the number of documents n. Yet, the dominating factor remains with the clustering, so the time complexity of the entire landscape generation process is $\boldsymbol{O(n \log(n))}$. This matches the performance of other scalable algorithms, such as [17], which however do not provide support for incremental layout computation. While we have experienced with data set sizes up to 20.000 documents, we still need to conduct large-scale experiments to make reliable statements with respect to scalability.

4 Evaluation

For evaluating the incremental computation framework we computed ten consecutive landscapes for 2000 documents from the environmental blog data set of the *Media Watch on Climate Change* [16], a Web content aggregator about climate change and related environmental issues that serves static versions of the information landscapes presented in this paper as part of a multiple coordinated view representation. Each week new documents are gathered via the *webLyzard* crawler. For each incremental step, new documents replace an equal number of old documents from the set of 2000 documents. Each new incremental computation is based on the previous one. For comparison, we compute the landscape for the same document set in a non-incremental manner. The projection quality in both landscapes is then evaluated using stress values [21].

However, the stress value computation requires the computation of distances (dissimilarities) for all pairs of documents in high-dimensional space, which is quadratic in time. To speed up the process and to be capable of handling large data sets, we introduce a faster variant which approximates the true stress values. We used geometric mean of the similarity of one document with the centroid of the second document and the similarity of second document with the centroid of the first document as an estimated value of similarity between both documents. All measurements were performed on a 2.66GHz Intel Xeon X5355 CPU with 8GB of memory, running 64-bit versions of Linux and Java v1.6.0_29.

The resulting stress values for both computation types are summarized in Fig. 3 (left). The initial sample of 2000 documents was taken from September 30th, 2011. Every week this document selection changes, i.e. new documents arrive whereas the same number of documents, the oldest ones, are removed resulting in a set of constant size. Stress values for both computation types are decreasing while values for the incremental computation appear to be slightly lower than for the non-incremental computation. In the non-incremental case, the curve exhibits more fluctuations, e.g. the peak on November 4th. In our opinion this behavior is due to k-means' and FDP's sensitivity to initial conditions. We hypothesize that stress values for the incremental computations are lower because these weekly incremental changes have the potential to shake the FDP process out of local minima so that the performance can improve. The experimental results corroborate that our algorithmic approach is capable of accurately generating dynamic information landscapes in an incremental manner.

To examine the algorithm's execution times for different data set sizes, we experimentally verified the runtime estimates for individual processing steps given in Sections 3.1 to 3.3. Fig. 3 (right) summarizes timing results of landscape computations for eight different document set sizes ranging from 2000 to 16000. Processing steps include document-term matrix preparation (A), clustering (B), document positioning (C) (including cluster positioning with FDP which is in constant time for fixed number of clusters) and peak detection, label positioning and image construction (D). Graph (T) reflects the total runtime for generating dynamic topography information landscapes for different data set sizes. According to Fig. 3 (right), the clustering step (B) appears to be the algorithm's runtime bottleneck.

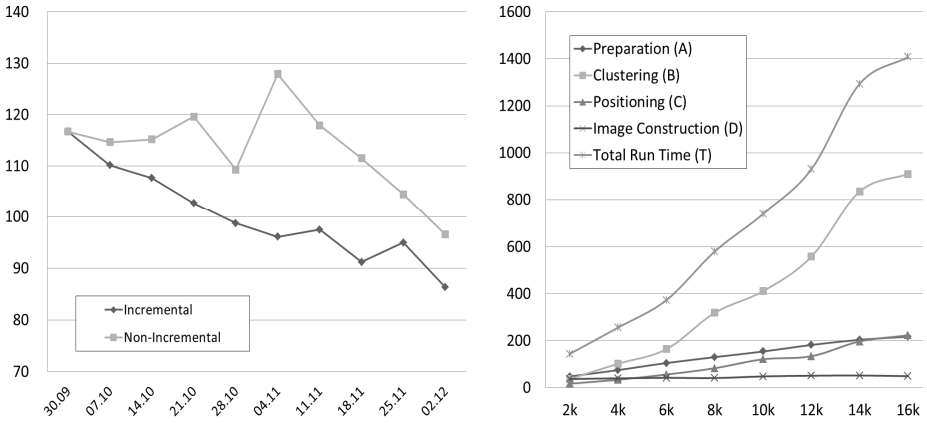

Fig. 3. Left: The stress values (y-axis) for incrementally computed documents layout and for non-incrementally computed documents layout over a period of 10 weeks; Right: Run times in seconds (y-axis) for landscape computation framework with different document sets (x-axis)

5 Conclusion

We have introduced and evaluated an incremental approach to generating dynamic topography information landscapes, and applied this approach to visualize the content dynamics of environmental blogs. Our method combines well-known algorithmic approaches, such as k-means clustering and force-directed placement, and introduces an improved method for fast document positioning which relies on previously computed cluster centroid positions. In experiments, we have compared the quality of incrementally and non-incrementally computed layouts where the incremental version achieves not only comparable, but even slightly superior stress values.

By capturing changes in textual data repositories such as news and social media archives, and by revealing the emergence and decay of major topics in such repositories, an incremental version for computing information landscapes extends the repertoire of existing Web intelligence and social media analytics applications such as the *Media Watch on Climate Change* (www.ecoresearch.net/climate).

Although some of incremental ideas are discussed in [27, 28, 29, 30], this paper contributes by presenting a novel document positioning method and evaluates document positioning improvements on subsequent incremental landscape computations.

Future work will focus on improving layout quality by utilizing semantic information in the process of calculating similarities between documents. These semantics will help us to better handle linguistic concepts such as synonymy and thus to capture more implicit, meaningful associations amongst textual resources.

Acknowledgement. The work presented in this paper was developed within the DIVINE project (www.weblyzard.com/divine), funded by the Austrian Ministry of Transport, Innovation & Technology (BMVIT) and the Austrian Research Promotion

Agency (FFG) within the strategic objective FIT-IT (www.ffg.at/fit-it). The Know-Center is funded within the Austrian COMET Program (Competence Centers for Excellent Technologies) under the auspices of BMVIT, the Austrian Ministry of Economics and Labor, and by the State of Styria.

References

1. Allan, J., Carbonell, J., Doddington, G., Yamron, J., Yang, Y.: Topic Detection and Tracking. Pilot Study Final Report (1998)
2. Artac, M., Jogan, M., Leonardis, A.: Incremental PCA for on-line visual learning and recognition. In: Proceedings of the 16th International Conference on Pattern Recognition, pp. 781–784 (2002)
3. Arthur, D., Vassilvitskii, S.: K-means++: The advantages of careful seeding. In: Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 1027–1035 (2007)
4. Basalaj, W.: Incremental multidimensional scaling method for database visualization. In: Proceedings of SPIE - The International Society for Optical Engineering, pp. 149–158 (1999)
5. Brand, M.: Incremental Singular Value Decomposition of Uncertain Data with Missing Values. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002, Part I. LNCS, vol. 2350, pp. 707–720. Springer, Heidelberg (2002)
6. Christensen, J., Marks, J., Shieber, S.: An empirical study of algorithms for point feature label placement. ACM Trans. on Graphics 14(3), 203–232 (1995)
7. Dhillon, I.S., Modha, D.S.: Concept decompositions for large sparse text data using clustering. Machine Learning 42(1/2), 143–175 (2001)
8. Ester, M., Kriegel, H.-P., Sander, J., Wimmer, M., Xu, X.: Incremental clustering for mining in a data warehousing environment. In: Proceedings of 24th International Conference on Very Large Data Bases (VLDB 1998), pp. 323–333 (1998)
9. Fisher, D.: Knowledge acquisition via incremental conceptual clustering. Machine Learning 2, 139–172 (1987)
10. Fruchterman, T., Reingold, E.: Graph drawing by force-directed placement. Software - Practice and Experience 21, 1129–1164 (1991)
11. Gennari, J., Langley, P., Fisher, D.: Models of incremental concept formation. Artificial Intelligence 40, 11–61 (1989)
12. Hartigan, J.A.: Clustering Algorithms. John Wiley and Sons, Inc., New York (1975)
13. Havre, S., Hetzler, E., Whitney, P., Nowell, L.: ThemeRiver: Visualizing thematic changes in large document collections. IEEE Transactions on Visualization & Computer Graphics 8(1), 9–20 (2002)
14. Heaps, H.H.: Information Retrieval: Computational and Theoretical Aspects, pp. 206–208. Academic Press (1978)
15. van Herk, M.: A fast algorithm for local minimum and maximum filters on rectangular and octagonal kernels. Pattern Recognition Letters 13(7), 517–521 (1992)
16. Hubmann-Haidvogel, A., Scharl, A., Weichselbraun, A.: Multiple coordinated views for searching and navigating web content repositories. Information Sciences 179(12), 1813–1821 (2009)
17. Jourdan, F., Melancon, G.: Multiscale hybrid MDS. In: Proceedings of the Eighth International Conference on Information Visualisation (IV 2004), pp. 388–393 (2004)

18. Kanerva, P., Kristofersson, J., Holst, A.: Random indexing of text samples for latent semantic analysis. In: Proceedings of the 22nd Conference of the Cognitive Science Society, pp. 103–106 (2000)
19. Kouropteva, O., Okun, O., Pietikäinen, M.: Incremental locally linear embedding. Pattern Recognition, 1764–1767 (2005)
20. Krishnan, M., Bohn, S., Cowley, W., Crow, V., Nieplocha, J.: Scalable visual analytics of massive textual datasets. In: 21st IEEE Int'l Parallel and Distributed Processing Symposium 2007. IEEE Computer Society (2007)
21. Kruskal, J.: Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. Psychometrika 29(1), 1–27 (1964)
22. Muhr, M., Granitzer, M.: Automatic cluster number selection using a split and merge k-means approach. In: Proceedings of the 20th International Workshop on Database and Expert Systems Application, pp. 363–367 (2009)
23. Pang, S., Ozawa, S., Kasabov, N.: Incremental linear discriminant analysis for classification of data streams. IEEE Transactions on Systems Man and Cybernetics 35, 905–914 (2005)
24. Pelleg, D., Moore, A.: X-means: Extending k-means with efficient estimation of the number of clusters. In: Proceedings of the 17th International Conference on Machine Learning, pp. 727–734 (2000)
25. Razaz, M., Hagyard, D.M.P.: Efficient convolution based algorithms for erosion and dilation. In: Proc. of the IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing (NSIP 1999), pp. 360–363 (1999)
26. Ribert, A., Ennaji, A., Lecourtier, Y.: An incremental hierarchical clustering. In: Proceedings of the Vision Interface Conference, pp. 586–591 (1999)
27. Sabol, V., Kienreich, W.: Visualizing Temporal Changes in Information Landscapes. Poster and Demo at the EuroVis (2009)
28. Sabol, V., Scharl, A.: Visualizing Temporal-Semantic Relations in Dynamic Information Landscapes. In: 11th International Conference on Geographic Information Science, Semantic Web Meets Geospatial Applications Workshop. AGILE, Girona (2008)
29. Sabol, V., Syed, K.A.A., Scharl, A., Muhr, M., Hubmann-Haidvogel, A.: Incremental Computation of Information Landscapes for Dynamic Web Interfaces. In: Proc. of the 10th Brazilian Symposium on Human Factors in Computer Systems, pp. 205–208 (2010)
30. Sabol, V., Kienreich, W., Muhr, M., Klieber, W., Granitzer, M.: Visual Knowledge Discovery in Dynamic Enterprise Text Repositories. In: Proceedings of the 13th International Conference on Information Visualisation (IV 2009). IEEE Computer Society (2009)
31. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Incremental singular value decomposition algorithms for highly scalable recommender systems. In: Proceedings of the 5th International Conference on Computer and Information Science, pp. 399–404 (2002)
32. Scharl, A., Weichselbraun, A., Liu, W.: Tracking and modelling information diffusion across interactive online media. International Journal of Metadata, Semantics and Ontologies 2(2), 135–145 (2007)
33. Slagle, J.R., Chang, C.L., Heller, S.R.: A clustering and data-reorganizing algorithm. IEEE Trans. Syst. Man Cybern. 5, 125–128 (1975)
34. Yan, J., Cheng, Q., Yang, Q., Zhang, B.: An Incremental Subspace Learning Algorithm to Categorize Large Scale Text Data. In: Zhang, Y., Tanaka, K., Yu, J.X., Wang, S., Li, M. (eds.) APWeb 2005. LNCS, vol. 3399, pp. 52–63. Springer, Heidelberg (2005)
35. Zhang, T., Ramakrishnan, R., Livny, M.: BIRCH: an efficient data clustering method for very large databases. In: Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data, pp. 103–114. ACM, New York (1996)