Multidimensional Projection Applied to Textual Search Results Visualization

Erick Gomez-Nieto
ICMC/SME
Universidade de São Paulo
São Carlos, SP, Brazil
Email: erick.gomez@ucsp.pe

Luis Gustavo Nonato
ICMC/SME
Universidade de São Paulo
São Carlos, SP, Brazil
Email: gnonato@icmc.usp.br

***Abstract*—Internet users are very familiar with the results of a search query displayed as a ranked list of snippets. Each textual snippet shows a content summary of the referred web page and a link to it. This display has many advantages, e.g., it affords easy navigation and is straightforward to interpret. Nonetheless, any user of search engines could possibly report some experience of disappointment with this metaphor. Indeed, it has limitations in particular situations, as it fails to provide an overview of the document collection retrieved. Several search tasks would be easier if users were shown an overview of the returned documents, organized so as to reflect how related they are, content-wise. We propose a visualization technique to display the results of web queries aimed at overcoming such limitations. It combines the neighborhood preservation capability of multidimensional projections with the familiar snippet-based representation by employing a multidimensional projection to derive two-dimensional layouts of the query search results that preserve text similarity relations, or neighborhoods. If the snippets are displayed directly according to the derived layout they will overlap considerably, producing a poor visualization. We overcome this problem by defining an energy functional that considers both the overlapping amongst snippets and the preservation of the neighborhood structure as given in the projected layout. The resulting visualization conveys both a global view of the query results and visual groupings that reflect related results*.**

***Keywords*-Information Visualization; Visual Data Mining; Multidimensional Projection; Web Search Visualization**

I. Introduction

Searching for information on the Internet is routine task to millions of users. The typical procedure consists in providing textual queries to a search engine, which returns a ranked list of textual snippets each containing a content summary and a link to the referred document (or web page). A ranked list of snippets is quite simple, straightforward to interpret, and it turns out to be effective in focused search tasks that require locating a particular web page or document [1]. Nonetheless, it also has limitations likely to hamper user experience when exploring and analyzing search results in other scenarios. In fact, a ranked list fails to provide an overview of the collection of retrieved documents, making it difficult and time consuming to figure out how documents are related contentwise. For example, if a user queries Google's search engine on the keywords "jaguar features", the first returned page lists snippets on at least four distinct subjects, namely, the animal, the car brand, a fan club of old Jaguar cars, and a video game console.

Information visualization tools offer users more flexible mechanisms for inspecting and navigating the result of textual queries. Some existing methods preserve the snippet list paradigm while enhancing it with visual resources such as color glyphs and tag clouds, attempting to convey more information on the contents of the returned documents. Although interesting and potentially useful, those visual resources reveal no information on the structure of document neighborhoods, that is, which documents share similar content and how many different subjects appear in the search results. Other classes of methods depart from the ranked list paradigm and come up with alternatives such as thumbnails and dimensionality reduction, which favor better understanding of document content and the identification of groups of similar documents. However, those methods tend to be visually more intricate, demanding from users greater effort to locate and inspect specific documents. Moreover, current methods process the full content of each document *a priori* to create a visualization, therefore incurring in an added computational cost that prevents their easy plugging into standard search engines.

The technique introduced in this work is effective for visualizing textual query results whilst overcoming the drawbacks just discussed. It exploits the intrinsic neighborhood preservation nature of multidimensional projections to obtain a snippet-based two-dimensional layout that preserves the simplicity and usability of textual snippets while emphasizing content related groups of documents. The unique combination of a similarity-based layout with the textual snippets brings out a powerful mechanism to organize and present textual search results that retains the familiar snippet paradigm, thus avoiding complex interfaces and visual metaphors. Moreover, the visualization is created only from the information in the textual snippets, rendering the proposed method computationally efficient and easy to plug into conventional search engines.

The proposed method relies on an energy functional that considers both the overlapping between snippets and the neighborhood structure of returned documents. The minimum

*M.Sc. dissertation at ICMC/USP. Available at http://sites.google.com/site/erickgomeznieto/publications *(In Portuguese)*

energy of such functional provides a neighborhood preserving two-dimensional arrangement of the textual snippets with minimum overlap. Snippet size can change according to properties such as document rank and colors can be employed to highlight clusters of similar documents.

Contributions:

- *2D Snippet Layout:* A new method, which we call *ProjSnippet*, to display the results of textual queries that integrates textual snippets and a multidimensional projection layout into a simple and intuitive visualization.
- *Energy Functional:* The *ProjSnippet* layout relies on a new overlap removal energy functional that considers both the neighborhood relations between snippets and their overlapping in the visual space.

II. ProjSnippet

The proposed technique comprises three steps as shown in the pipeline in Fig. 1: pre-processing of search results, multidimensional projection, and optimization. In the first step each entry returned from a textual query is processed and its term frequency vector extracted (see [2] for details on term frequency extraction). Stemming and stopword removal are applied and Luhn's lower and upper cuts established to compute the *tf-idf* vector representation of each snippet. Only the summary texts contained in the hits are processed, rather than the full content of the referred documents or web pages, which renders the visualization algorithm fast. This is a important requirement, e.g. for interactivity and for plugging the solution into standard browsers.

Each term frequency vector may be handled as a point in a high-dimensional space that can be mapped to the visual space with a multidimensional projection technique. Albeit our current implementation adopts the *Least Squares Projection* (LSP) [3] – due to its good accuracy in distance preservation and low computational cost – any projection technique with similar properties might be employed (see [4] for an up-to-date survey of efficient multidimensional projection methods). The projection preserves much of the neighborhood structure of the original data, ensuring that similar instances are placed close to each other in the visual space.

The following step is to embed the content of each snippet within a rectangle whose bottom left corner is placed in the snippet's projected position. A rectangle's height and width are settled according to the rank of its corresponding snippet in the retrieved document list, so that higher ranked snippets are assigned larger rectangles. The k-means++ algorithm is applied to the projected layout to identify clusters of similar documents (taking as metric the Euclidean distance in the visual space), and rectangles in the same cluster may be assigned the same color to highlight groups.

A major drawback at this stage of the pipeline is that rectangles enclosing the snippets overlap considerably, impairing identification of individual entries and the perception of the document neighborhood structure. The final step (rightmost box in Fig. 1) optimizes the placement of the snippets so as to avoid overlapping while preserving data neighborhoods as computed by the projection. The optimization is driven by an energy functional, detailed next.

A. The Energy Functional

The energy functional E comprises two components, one that takes into account the overlap of snippets, denoted by E_O, and a second component related to the neighborhood relations resulting from the projection step, denoted by E_N. In mathematical terms, the energy E is written as:

$$E = (1-\alpha)E_O + \alpha E_N \tag{1}$$

where the parameter $\alpha \in [0,1]$ balances the relative contributions of both E_O and E_N in the total energy.

Energy E, as well as E_O and E_N, are functions of the coordinates of the bottom-left corners of the rectangles embedding the snippets, which initially correspond to the projected coordinates of the high-dimensional snippet vectors.

Overlapping Energy Aiming at enhancing overall visibility and readability of the visualization, the energy E_O must be defined so as to minimize the overlap/intersection of nearby snippets. This is achieved with a function that measures the distance between the left corners of the rectangles. This function is smooth, attains its minimum value when no overlapping takes place and takes higher values when rectangle overlap is greater. Smoothness is an important property here, as it allows resorting to simple and efficient optimization methods which are mandatory for quick generation of the final visualization.

Let $\vec{x},\vec{y} \in \mathbf{R}^n$ be the coordinate vectors of the bottom left corner of each rectangle and $\vec{v},\vec{h} \in \mathbf{R}^n$ be vectors whose components are the vertical and horizontal dimensions of each rectangle. We first define two auxiliary functions to simplify the presentation:

$$[x]_+ = \begin{cases} x & x \geq 0 \\ 0 & x < 0 \end{cases}$$

and

$$O_{i,j}(\vec{x},\vec{h}) = \begin{cases} \frac{1}{h_j^4}[h_j^2 - (x_i - x_j)^2]_+^2 & x_i \geq x_j \\ \frac{1}{h_i^4}[h_i^2 - (x_i - x_j)^2]_+^2 & x_i < x_j \end{cases}.$$

where x_i, h_i and x_j, h_j denote, respectively, the x-coordinates of the bottom left corner and the lengths of rectangles i and j. Notice that $O_{i,j}(\vec{x},\vec{h})$ is zero when there is no horizontal overlapping of rectangles i and j and attains its maximum value of 1 when the x-coordinate of the left corners of both rectangles coincide. Function $O_{i,j}$ works similarly if y-coordinates and heights are used as arguments, i.e., $O_{i,j}(\vec{y},\vec{v})$.

From definitions above we set E_O as:

$$E_O = \frac{2}{n(n+1)} \sum_{i=1}^{n} \sum_{j=i+1}^{n} \left[O_{i,j}(\vec{x},\vec{h}) O_{i,j}(\vec{y},\vec{v}) \right]. \tag{2}$$

where n is the number of projected points. The definition of $O_{i,j}$ clearly guarantees that E_O is continuously differentiable and it ranges in the interval $[0,1]$.

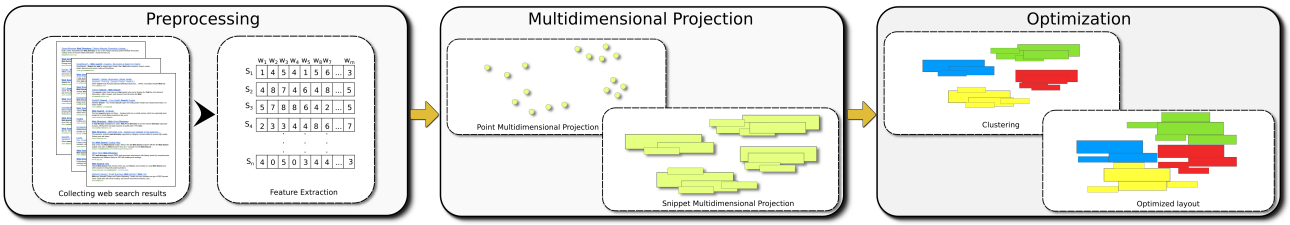
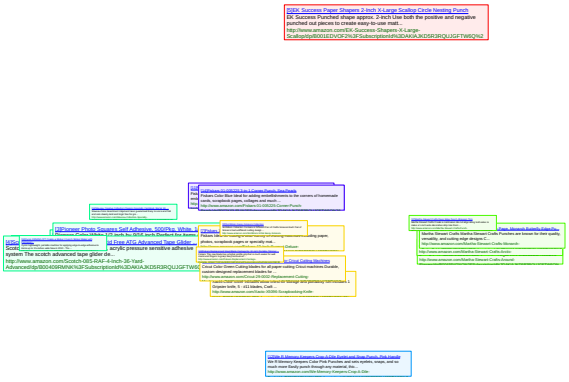
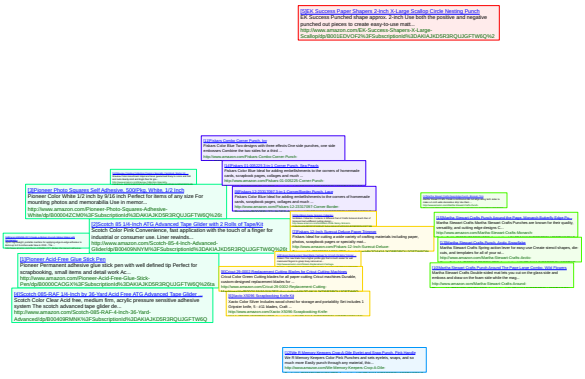


Fig. 1. Pipeline to generate the neighborhood-preserving snippet visualization. Snippet textual content is processed and represented as high-dimensional data points (left). The high-dimensional data is mapped to the visual space and snippets are embedded into rectangles (middle). Optimization is applied to avoid overlap while preserving neighborhoods.



(a) Without optimization



(b) $E = \frac{1}{2}(E_O + E_N)$

Fig. 2. a) Layout without optimization; b) both energies E_O and E_N combined ($\alpha = 0.5$).

Neighborhood Energy The minimization of E_O spreads textual snippets in the visual space so as to prevent rectangles from overlapping. However, this minimization process is likely to spoil the neighborhood structure established by the multidimensional projection, placing similar snippets far apart in the final visualization.

The energy term E_N is introduced to balance the effect of the overlapping energy during optimization. In practice, the energy E_N is defined from a k-nearest-neighbor graph G constructed from the projected "snippet-vectors" (our implementation uses $k = 10$). In order to ensure G is connected, any disconnected components resulting from constructing the k-nearest-neighbor graph are connected by adding to G the shortest edge between them.

Let L be the $n \times n$ matrix with entries l_{ij} given by:

$$l_{ij} = \begin{cases} -1/|i| & \text{if } j \neq i \text{ and } \overline{ij} \text{ is an edge of G} \\ 1 & \text{if } j = i \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

where $|i|$ is the valence of node i.

Denoting by $\vec{x}^0$ and $\vec{y}^0$ the x and y coordinate vectors of the nodes of G (recall that $\vec{x}^0$ and $\vec{y}^0$ result from the multidimensional projection step) we define the differential vectors $\vec{\delta}_x$ and $\vec{\delta}_y$ as:

$$\vec{\delta}_x = L\vec{x}^0, \qquad \vec{\delta}_y = L\vec{y}^0. \quad (4)$$

Notice that the components of $\vec{\delta}_x$ and $\vec{\delta}_y$ are given, respectively, by:

$$x_i^0 - \frac{1}{|N_i|}\sum_{j \in N_i} x_j^0, \qquad y_i^0 - \frac{1}{|N_i|}\sum_{j \in N_i} y_j^0 \quad (5)$$

In less mathematical terms, $\vec{\delta}_x$ and $\vec{\delta}_y$ measure how much each node deviates from the average of its neighbors. Therefore, we define the neighborhood energy as

$$E_N = \frac{n^2}{2\left(\left\|\vec{\delta}_x\right\|^2 + \left\|\vec{\delta}_y\right\|^2\right)}\left(\left\|L\vec{x} - w\vec{\delta}_x\right\|^2 + \left\|L\vec{y} - w\vec{\delta}_y\right\|^2\right). \quad (6)$$

It is not difficult to realize that E_N will be minimal when $\vec{x}$ and $\vec{y}$ are such that their differentials $L\vec{x}$ and $L\vec{y}$ are proportional to the initial differentials $\vec{\delta}_x$ and $\vec{\delta}_y$. In other words, the energy term E_N is minimized when neighborhood relations are preserved during optimization. The unknown w is added to the optimization to ensure that any scale of the points is a minimum of the neighborhood energy (w is optimized together with $\vec{\delta}_x$ and $\vec{\delta}_y$).

The normalization factor $\frac{n^2}{2}\left(\left\|\vec{\delta}_x\right\|_2^2 + \left\|\vec{\delta}_y\right\|_2^2\right)^{-1}$ is necessary to ensure that the range of E_N is in the same order of magnitude as E_O, so that both terms play similar roles (controlled by the parameter α) in the total energy E.

Fig. 2 illustrates the result of performing the optimization on the projected snippets shown in Fig. 2a. Fig. 2b shows the outcome of the optimization procedure with both energy terms equally balanced.

B. Computational Aspects and Implementation Details

Bounds related to the sizes of the visualization windows are imposed as constraints for the minimization of the energy (1).

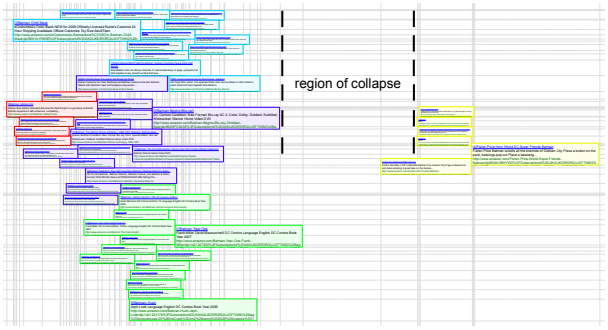
This is necessary since for a sufficiently large positive number K, the coordinate vectors $\vec{x} = K\vec{x}^0$ and $\vec{y} = K\vec{y}^0$, $(w = K)$ correspond to a global minimizer of E, as no overlap should happen and differentials are preserved by properly scaling the layout. However, the minimal solution given by scaling is prone to spread the snippets far apart, resulting in unpleasant and useless visualizations.

Therefore, denoting the horizontal and vertical bounds of the visualization window by x_{min}, x_{max} and y_{min}, y_{max} the minimization problem becomes:

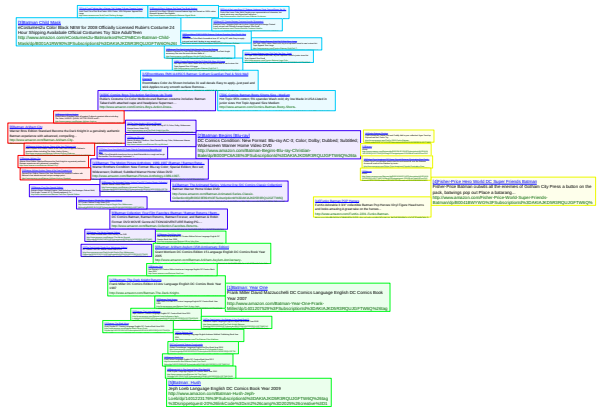
$$\begin{aligned} \min \quad & (1-\alpha)E_O + \alpha E_N \\ \text{s.t.:} \quad & x_{min} \le x_i \le x_{max} - h_i, \quad i = 1,\ldots,n \\ & y_{min} \le y_i \le y_{max} - v_i, \quad i = 1,\ldots,n. \end{aligned} \tag{7}$$

which ensures that all rectangles lie within the visualization window, therefore preventing an exaggerate scaling effect.

The minimization is accomplished by a globally convergent local optimization method, namely the Method of Moving Asymptotes, available from the *NLopt* library.



(a) Rectilinear grid to perform carving.



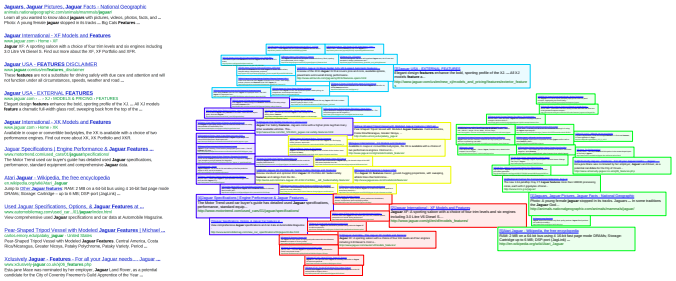
(b) With carving strategy.

Fig. 3. Reducing white space with a carving mechanism.

Reducing White Space In order to reduce white space in the final layout we implemented a simplified version of the *seam carving strategy* [5]. The idea is to partition the snippet-free regions of the visualization window into a rectilinear grid, as illustrated in Fig. 3a. Seams are then created by collapsing rectangular grid cells from left to right and then from top to down. A cell is collapsed if and only if all the snippets in the clusters affected by the collapse can be moved horizontally or vertically. If only part of a cluster can be moved no collapsing is performed. Such a simple carving mechanism runs quickly and it obviously preserves the clusters, resulting in pleasant layouts (see Fig. 3b). There are more sophisticated carving strategies able to further remove white space, but they are computationally costly and tend to mess neighborhoods.

III. Results, Comparisons and Evaluation

In the following we present examples illustrating the ProjSnippet visualization and its capability to globally convey the results of a web query while emphasizing related hits in a meaningful way. A k-means++ clustering has been applied just to color the rectangles to visually highlight groups of similar snippets and provide some visual segregation to facilitate user inspection. The optimization procedure has been performed with $\alpha = 0.3$ (our default value).



(a) First page of Google (b) ProjSnippet showing 64 snippets (more than six Google's pages)

Fig. 4. Google view and ProjSnippet view of the results of a query with terms "jaguar features".

The first example illustrates a visualization displaying the results of a query on the terms "jaguar features" submitted to Google's search engine. The view in Fig. 4a shows the first page from Google, listing the 10 best ranked snippets. Fig. 4b displays a ProjSnippet view with the 64 best ranked snippets, equivalent in content to more than six Google pages. Inspecting this visualization one notices that the snippets on the left (cyan, red, blue, yellow) all refer to different models of Jaguar cars, whereas the green ones on the right refer to a surprising variety of topics. Those include multiple references to the wild animal (3 snippets) and also to supercomputer models named Jaguar (2 instances). There are also unique references to an earlier MacOs operating system named Jaguar, to a video game, a swimming pool brand, a hair product brand, an aircraft model and a few other varied stuff. Looking at the snippets in the left area, one identifies that most snippets in the blue cluster contain general references to the car brand, whereas the each of the three other clusters refer mostly to a specific Jaguar model, namely most yellow snippets refer to the XK model, cyan snippets refer to XJ and red to XF models. There are some noticeable exceptions, e.g., a yellow snippet refers to the XF model and a blue one refers to the XJ model. Still, overall the final layout does a very good job of depicting a global view of the search hits, as

far grouping/separating similar/dissimilar results is concerned. Notice that it is pretty difficult to handle such a variety topics and subtopics in Google's list-based view, which indeed brings only cars, animals and the game as the topics in the first page. Users can interact with the visualizations to navigate directly from the snippets, e.g., to further inspect page contents, as afforded by the conventional list-based metaphor.

The effect of varying the parameter α (in equation (1)) is illustrated in Fig. 5, which shows optimized layouts (without applying the carving mechanism) of the results from a query on terms "wave applications" posed to Bing's search engine. In Fig. 5a only the overlapping energy has been considered ($\alpha = 0$). There is no overlapping, but neighborhoods are clearly not preserved and snippets are far too spread. Inspecting Fig. 5(b, c, d) one observes how similar snippets get more tightly connect as α values increase. Notice that even with large values of α ($\alpha = 0.8$ in Fig. 5d) the snippets do not overlap unduly, showing the robustness of ProjSnippet.

A. Comparing with Overlapping Removal Heuristics

In order to assess the effectiveness of the overlap removal mechanism built into ProjSnippet we have compared it with four well-known heuristics that avoid overlapping while still preserving the semantic relations as much as possible, namely, VPSC, PRISM, Voronoi-based and RWordle-C (a brief review of these can be inspected in [6]), regarding the following metrics:

Euclidean distance: this metric measures how much the boxes move during the overlap removal process. Less movement is preferred, since the original configuration is better preserved.
Layout similarity: this metric attempts to quantify how much neighborhood structures are affected by the overlap removal mechanism and it is derived from the Frobenius metric.
Size increase: determines the relative changes in size as well as the compactness of the representation.
Neighborhood preservation: it computes the average percentage of preservation of the k-nearest neighbors of each box in the final layout.

Fig. 6 shows the layouts produced by the algorithms when applied to some different data sets (extracted from Amazon, Google and Bing web search engines). One notices that ProjSnippet achieves more pleasing visual results, as compared with existing heuristics. On a first glance, its layouts resemble those obtained by RWordle-C, but ProjSnippet is more effective to maintain similar elements grouped together (notice, for example, the red group for "Jaguar Features").

Fig. 7 summarizes the quantitative results of the above metrics. Notice that ProjSnippet performed quite well, resulting in better values than the other methods, in most cases.

Fig. 8 shows a plot that considers all metrics simultaneously. Each overlap removal technique has been represented as a four-dimensinal vector $(E_a, \sigma_a, S_a, k_a)$, where E_a, σ_a, S_a, k_a are the average values of the metrics E, σ, S, and k-nearest neighbors computed for each technique, over all data sets. The points denoted by "best", "average", and "worst" in Fig. 8

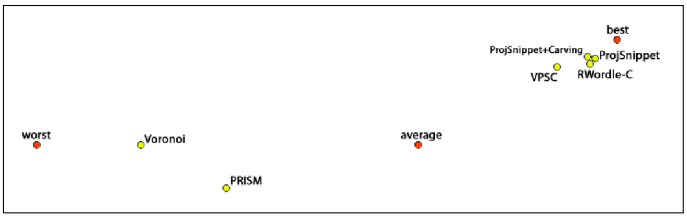


Fig. 8. Comparison of overlap removal methods regarding the four metrics simultaneously.

were created artificially as four dimensional vectors containing the better, average and worst results computed taking account all methods. The multidimensional projection method LAMP [4] was used to project the four dimensional data to the visual space. The technique closer to "best" is the one that, on average, has shown the best performance, relative to all metrics. One can clearly sees that ProjSnippet is the technique closer to the best result.

IV. Conclusion

We introduced ProjSnippet, a novel technique to visualize the collection of textual snippets returned from a web query. The method builds intuitive and meaningful layouts that optimize the placement of snippets by employing an innovative energy functional that considers both overlapping removal and preservation of neighborhood structures.

We showed results illustrating how the ProjSnippet layouts convey a global view of the results from a query while allowing for identifying similar content through a clustering mechanism. Since ProjSnippet relies only on information extracted from the textual snippets, it can be plugged into search engines in a straightforward manner, with a modest impact on the computational times. The unique combination of simplicity, low computational cost, and flexibility renders ProjSnippet an attractive alternative for visualizing web queries results. We are currently investigating interactive mechanisms to enable a free navigation in the snippet-based layout.

References

[1] J. Teevan, E. Cutrell, D. Fisher, S. M. Drucker, G. Ramos, P. AndrÃl', and C. Hu, "Visual snippets: summarizing web pages for search and revisitation." in *CHI'09*, 2009, pp. 2023–2032.

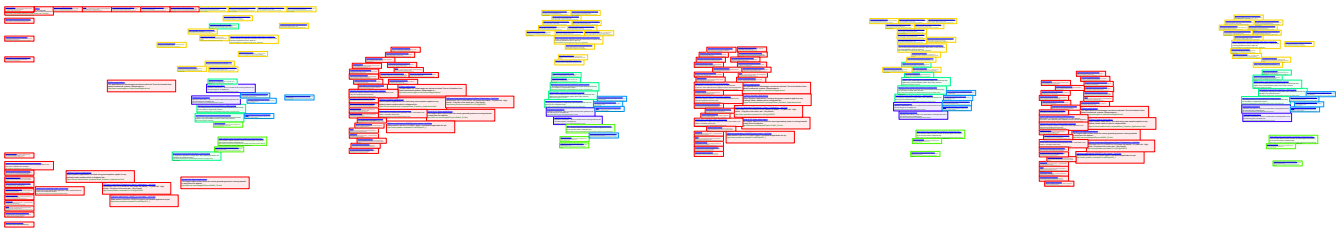
[2] G. Salton, "Developments in automatic text retrieval," *Science*, vol. 253, pp. 974–980, 1991.

[3] F. Paulovich, L. Nonato, R. Minghim, and H. Levkowitz, "Least square projection: A fast high-precision multidimensional projection technique and its application to document mapping," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 14, no. 3, pp. 564 –575, may-june 2008.

[4] P. Joia, D. Coimbra, J. Cuminato, F. Paulovich, and L. Nonato, "Local affine multidimensional projection," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, pp. 2563–2571, Dec. 2011. [Online]. Available: http://dx.doi.org/10.1109/TVCG.2011.220

[5] Y. Wu, T. Provan, F. Wei, S. Liu, and K.-L. Ma, "Semantic-preserving word clouds by seam carving," *Comput. Graph. Forum*, vol. 30, no. 3, pp. 741–750, 2011.

[6] H. Strobelt, M. Spicker, A. Stoffel, D. Keim, and O. Deussen, "Rolled-out Wordles: A Heuristic Method for Overlap Removal of 2D Data Representatives," *Computer Graphics Forum*, vol. 31, no. 3, pp. 1135–1144, 2012. [Online]. Available: http://diglib.eg.org/EG/CGF/volume31/issue3/v31i3pp1135-1144.pdf



(a) $\alpha = 0.0$ (b) $\alpha = 0.2$ (c) $\alpha = 0.5$ (d) $\alpha = 0.8$

Fig. 5. Effect of varying parameter α.

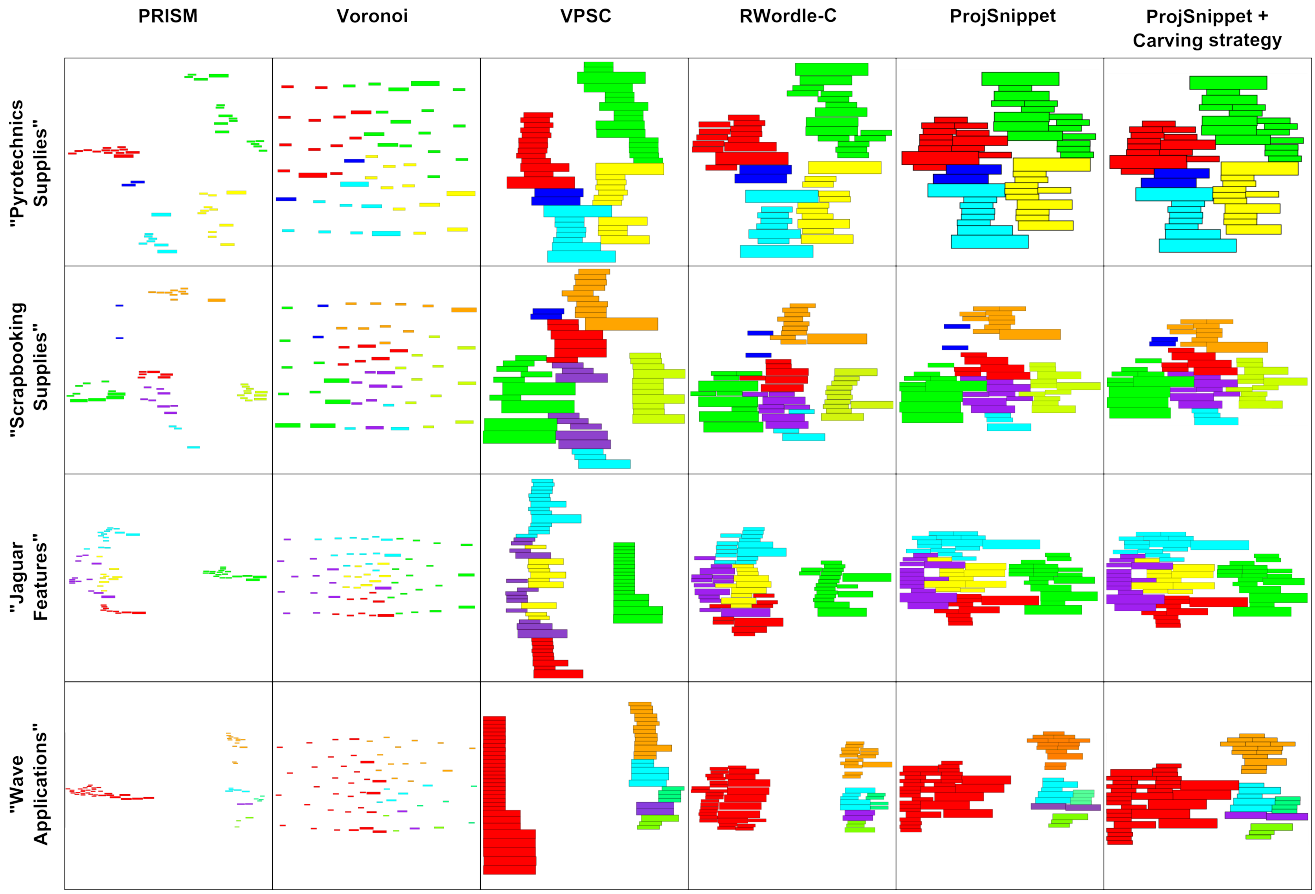


Fig. 6. Layouts produced by ProjSnippet, VPSC, PRISM, Voronoi-based, and RWordle-C for five distinct data sets.

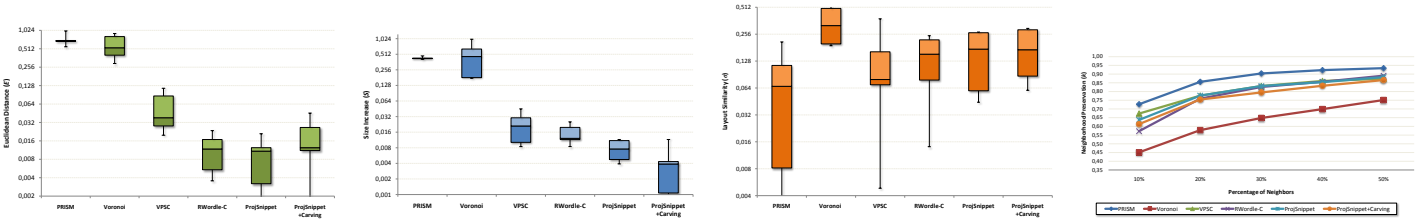


Fig. 7. Comparing ProjSnippet, VPSC, PRISM, Voronoi-based, and RWordle-C considering metrics Euclidean distance (a), Layout similarity (b), Size increasing, and Neighborhood preservation (d).