Generation of Semantic Layouts for Interactive Multidimensional Data Visualization

Erick Gomez-Nieto[*†] and Luis Gustavo Nonato[†]
[*]Research and Innovation Center in Computer Science
Universidad Católica San Pablo, Arequipa, Peru
[†]Instituto de Ciências Matemáticas e de Computação
Universidade de São Paulo
São Carlos-SP, Brazil

Abstract—Visualization methods make use of interactive graphical representations embedded on a display area in order to enable data exploration and analysis. These typically rely on geometric primitives for representing data or building more sophisticated representations to assist the visual analysis process. One of the most challenging tasks in this context is to determinate an optimal layout of these primitives which turns out to be effective and informative. Existing algorithms for building layouts from geometric primitives are typically designed to cope with requirements such as orthogonal alignment, overlap removal, optimal area usage, hierarchical organization, dynamic update among others. However, most techniques are able to tackle just a few of those requirements simultaneously, impairing their use and flexibility. In this dissertation, we propose a set of approaches for building layouts from geometric primitives that concurrently addresses a wider range of requirements. Relying on multidimensional projection and optimization formulations, our methods arrange geometric objects in the visual space so as to generate well-structured layouts that preserve the semantic relation among objects while still making an efficient use of display area. A comprehensive set of quantitative comparisons against existing methods for layout generation and applications on text, image, and video data set visualization prove the effectiveness of our approaches[1] [2].

I. INTRODUCTION

Arranging geometric primitives to generate meaningful layouts is a major task in visualization, which inherently appears in important applications such as word cloud constructionand visual boards.The difficulty in building layouts made up of dozens of geometric objects rests in the set of requirements to be handled simultaneously, e.g., readability, overlaps, object size, semantic proximity and area usage. Moreover, the number of data instances represented as geometric entities is typically much larger than the visualization area, demanding the use of clustering, hierarchies, and navigation resources to assist the visualization.

Although significant advances have been made towards building meaningful layouts from geometric primitives, existing techniques are formulated to deal with a limited number of requirements simultaneously, restricting their use to specific applications. For instance, techniques such as visual boards and small multiples provide well structured layouts which

are easily readable, but they pay the price of scalability. Hierarchical methods such as Treemaps mitigate the issue of scalability while making an efficient use of display area. However, readability and semantic organization of data are aspects not so easily handled by those methods. Overlap-free semantic preserving techniques generate somewhat structured layouts and keep instances with similar content close to each other. Nevertheless, they are not designed to make an efficient use of display area and also suffer from scalability.

Handling many requirements is not straightforward because distinct requirements can compete with each other during layout construction. For instance, to facilitate readability, layouts should be built with as large as possible geometric entities. However, large objects easily fill up the display area, thus limiting the number of instances that can be visualized. Therefore, finding an optimal balance among multiple concurrent requirements is a challenging task, which has not been completely tackled by existing methods.

In this dissertation we addressed this challenging problem by proposing new techniques for building layouts. We denominate to *Semantic Layout Arrangement* as the task of allocating efficiently a set of geometric instances, which summarizes a multidimensional dataset, into a fixed-size display area, subject to preserve, as much as possible and at all times, the semantic relationships among instances. The arrangement should simultaneously play with several requirements, such as area usage optimization, overlap removal, object scaling, orthogonal alignment and dynamic updating.

In the proposed methods, semantic relationships are established by a similarity measure. We use dimensionality reduction techniques for embedding data from multidimensional to visual space in order to, subsequently, build a map of geometric entities. Then, we applied the proposed optimization operators to rearrange entities according to the requirements we deem the most relevant, thereby generating meaningful visualizations.

This document presents a compilation of different techniques for interactive and semantic layouts generation for data visualization. Each proposed method brings new contributions for the field with the purpose of addressing and solving specific problems involved in generating geometric semantic layouts for interactive data visualization. Essentially, three

---

[1] Ph.D. thesis at ICMC-Universidade de São Paulo.
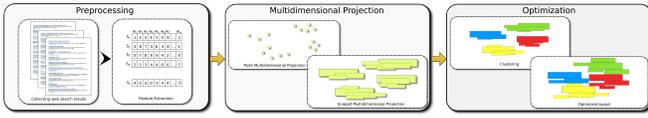[2] Email: emgomez@ucsp.pe, gnonato@icmc.usp.br

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

proposed methods (*ProjSnippet* [1], *MIOLA* [2] and *Dealing with Multiple Requirements* [3]) rely on novel optimization formulations for simultaneously dealing with requirements that were identified as the most relevant during our study. The last method (*Semantically Aware Dynamic Layouts* [4]) presented in this summary focuses on an application that demands semantic preserving layout updates during analyst's interactive data exploration.

To our knowledge, the proposed approaches figure among the most straightforward, efficient and intuitive alternatives to explore large amounts of multidimensional data while enabling a fluent interaction with data analysts.

II. PROJSNIPPET

The proposed technique comprises three steps as shown in the pipeline in Figure 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

Each term frequency vector may be handled as a point in a multidimensional space that can be mapped to the visual space with a multidimensional projection technique. Albeit our current implementation adopts the *Least Squares Projection* (LSP) [5] – due to its good accuracy in terms of distance preservation and low computational cost – any projection technique with similar properties might be employed. 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 (or its multidimensional data point) projected position. A rectangle's height and width are settled to reflect the rank of its corresponding snippet in the retrieved document list, so that better ranked snippets are assigned larger rectangles.

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 Figure 1) optimizes the placement of the snippets in order to avoid overlapping while preserving data neighborhoods as computed by the projection. The optimization is driven by an energy functional as follows.

A. *The energy functional*

The energy functional E involves two components, one that considers the overlap of snippets, denoted by E_O, and a second component related to the neighborhood relations resulting from the multidimensional 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 multidimensional snippet vectors. We omit the independent variables from the equations to simplify the notation.

The example illustrates a visualization displaying the results of a query on the terms "jaguar features" submitted to Google's search engine. The view in Figure 2a shows the 10 best ranked snippets shown in the first page. Figure 2b displays a ProjSnippet view with the 64 best ranked snippets. Inspection discloses 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, that 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 left region, 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 depicts a representative overview 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 results on cars, animals and the game in the first page.

III. MIXED INTEGER OPTIMIZATION FOR LAYOUT ARRANGEMENT (MIOLA)

In this section we describe a technique to tackle the problem of arranging rectangular boxes in the visual space so as to place objects representing similar content close to each other while avoiding overlaps. We formulate the problem as a *Mixed Integer Quadratic Programming Problem* (MIQP), which enables well structured layouts. In contrast to other optimal methods that take into account the similarity between instances, our approach does not rely on intersection tests, making the algorithm simpler to implement. Moreover, our technique is quite flexible, being able to generate different layouts by just handling optimization constraints.
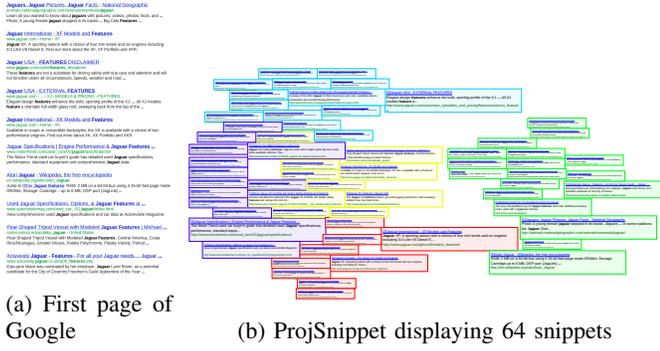
(a) First page of Google

(b) ProjSnippet displaying 64 snippets

Fig. 2: Google and ProjSnippet views of the results of a query with terms "jaguar features".

(a) Input layout.
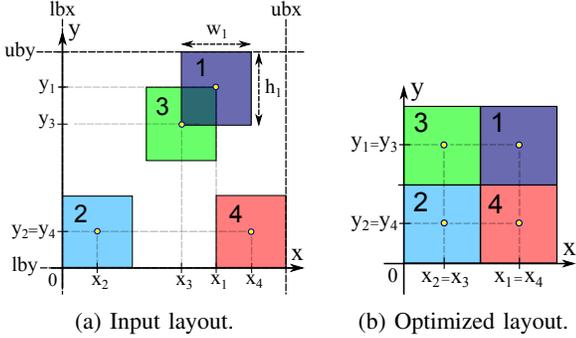
(b) Optimized layout.

Fig. 3: Inequalities (3) ensure that no boxes will not displayed offscreen while inequalities (4) preserve the order of the coordinates w.r.t. initial layout. (a) Input layout and (b) result after layout arrangement (orthogonal ordering: $x_{p_i} \leq x_{p_{i+1}}, p_1 = 2, p_2 = 3, p_3 = 1, p_4 = 4$; similarity for y case).

Let $B = \{B_1, B_2, ..., B_n\}$ be a set of n rectangular boxes arranged in the visual space such that the neighborhood structure of the boxes reflects a property of interest. For instance, if a data set is mapped to the visual space using a multidimensional projection technique and a box is centered on each projected data, the resulting arrangement makes neighbor boxes correspond to similar data. Boxes in this arrangement, however, should overlap considerably, impairing the visualization of individual boxes. In order to make each box visible, one has to displace the boxes in the two-dimensional space so as to remove overlaps, but preserving the initial neighborhood structures to keep similar objects close to each other. As described next, we formulate the problem above as a mixed integer quadratic programming optimization.

A. Problem statement

Let $B = \{B_1, B_2, ..., B_n\}$ be a set of n rectangular boxes initially positioned in a two-dimensional space. Each box B_i is specified by a four dimensional vector $B_i = (x_i, y_i, w_i, h_i) \in \mathbb{R}^4$, where (x_i, y_i), $w_i > 0$, $h_i > 0$ are the centroid, width and height of B_i, respectively (see Fig. 3(a)). Two boxes B_i and B_j do not overlap if and only if one of the following inequalities holds:

$$|x_j - x_i| \geq \frac{w_i + w_j}{2} \quad \text{or} \quad |y_j - y_i| \geq \frac{h_i + h_j}{2} \quad (2)$$

We refer to the inequalities in (2) as *non-overlap constraints*.

Moreover, the boxes must respect the bounds of the visualization window during displacement, that is,

$$\frac{w_i}{2} + lbx \leq x_i \leq ubx - \frac{w_i}{2} \text{ (x lower/upper bounds)}$$

$$\frac{h_i}{2} + lby \leq y_i \leq uby - \frac{h_i}{2} \text{ (y lower/upper bounds)} \quad (3)$$

where lbx, lby, ubx, uby are lower and upper visualization window bounds (constants) in x and y directions. If needed, visualization window bounds can also assume independent values for each box.

Equations (2) and (3) provide the conditions to be held so as to guarantee that boxes do not overlap and are inside a visualization window. However, those equations do not take into account neighborhood structures, thus neighbor boxes can be placed far apart from each other after the overlap removal process. One useful way to keep up the neighborhood relationships is to preserve the relative order of the centroids of boxes, that is,

$$x_{p_1} \leq x_{p_2} \leq ... \leq x_{p_n} \text{ (x orthogonal order)}$$
$$y_{q_1} \leq y_{q_2} \leq ... \leq y_{q_n} \text{ (y orthogonal order)} \quad (4)$$

where $p, q: \{1, 2, ..., n\} \rightarrow \{1, 2, ..., n\}$ are permutations of indices obtained by sorting the coordinates x and y of the centroids of boxes in the visual space (see Fig. 3).

Therefore, by moving the centroid of the boxes while ensuring Equations (2), (3), and (4) can generate an overlap free layout that preserves the initial neighborhood structures.

B. The MIQP formulation

The problem of positioning the boxes B_i in the visual space so as to ensure that Equations (2), (3), and (4) hold can be formulated as a *Mixed Integer Quadratic Programming Problem* as follows:

$$\min_z \quad f(\mathbf{z}) = \frac{1}{2}\mathbf{z}^T Q \mathbf{z} = \sum_{i=1}^{n} \sum_{j=i+1}^{n} Dist^2(B_i, B_j)$$

$$\text{subject to} \quad \begin{cases} A\mathbf{z} \leq \mathbf{b} \\ \mathbf{lb} \leq \mathbf{z} \leq \mathbf{ub} \end{cases} \quad (5)$$

$$\mathbf{z} = \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \\ \mathbf{r} \end{bmatrix}, \quad \begin{array}{l} \mathbf{x} = (x_1, x_2, ..., x_n)^\top \in \mathbf{R}^n \\ \mathbf{y} = (y_1, y_2, ..., y_n)^\top \in \mathbf{R}^n \end{array}$$

$$\mathbf{r} = (r_{12}, ..., r_{1n}, r_{23}, ..., r_{2n}, ..., r_{n-1n})^\top, r_{ij} \in \{0, 1\}$$

where $\mathbf{z}$ is the sought solution, $Dist(B_i, B_j)$ denotes the euclidian distance between (x_i, y_i) and (x_j, y_j), and vectors $\mathbf{lb}$ (lower bounds) and $\mathbf{ub}$ (upper bounds) are the visualization

window bounds as defined in the inequalities (3). Q is the positive semi-definite matrix composed by blocks L given by:

$$Q = \begin{pmatrix} \begin{bmatrix} L & 0 \\ 0 & L \end{bmatrix} & 0 \\ 0 & 0 \end{pmatrix}, \qquad L = nI_d - ones(n, n),$$

where I_d is the identity matrix and $ones(n, n)$ is the $n \times n$ matrix with all entries equal one.

Matrix A and vector $\mathbf{b}$ are defined so as to incorporate the constraints (2) and (4). Precisely, the ordering given by (4) allows us to write:

$$x_{p_1} \le x_{p_2} \le \dots \le x_{p_n} \Rightarrow x_{p_i} - x_{p_{i+1}} \le 0, \ i = 1 \dots n-1, \quad (6)$$

with a similar expression holding for y. Additionally, the sorting also allows us to get rid of the absolute value function in (2), that is,

$$x_{p_i} \le x_{p_j} \Rightarrow |x_{p_j} - x_{p_i}| = x_{p_j} - x_{p_i}. \text{ Thus,}$$

$$x_{p_i} - x_{p_j} \le -\frac{(w_{p_i} + w_{p_j})}{2}, \quad i < j, \quad (7)$$

with a similar expression holding for y. The variables r_{ij} allows for incorporating the *OR* condition defined in (2) into the optimization problem as follows:

$$x_{p_i} - x_{p_j} \le w_{ij} + M r_{ij} \Leftrightarrow y_{q_i} - y_{q_j} \le h_{ij} + M(1 - r_{ij}),$$

$$i < j, \quad r_{ij} \in \{0, 1\}, \quad (8)$$

where $w_{ij} = -\frac{(w_{p_i} + w_{p_j})}{2}$, $h_{ij} = -\frac{(h_{q_i} + h_{q_j})}{2}$ and M is a very large constant. The rationale behind the construction in Equation (8) is that if $r_{ij} = 0$, the constraint $x_{p_i} - x_{p_j} \le w_{ij}$ becomes mandatory while $y_{q_i} - y_{q_j} \le h_{ij} + M$ is naturally satisfied as M is a large number (if $r_{ij} = 1$, we have the opposite situation, instead). Therefore, optimizing the centroid positions and the decision variables r_{ij} simultaneously allows us to find the optimal position for the boxes while respecting all the constraints as stated in Equations (2), (3), and (4).

Equations (6) and (8) are incorporated into matrix A using the auxiliar matrices C_x, C_y, D_x, D_y as follows:

$$A = \begin{pmatrix} \begin{bmatrix} C_x & 0 & 0 \\ 0 & C_y & 0 \end{bmatrix} \\ \begin{bmatrix} D_x & 0 & -MI_d \\ 0 & D_y & MI_d \end{bmatrix} \end{pmatrix} \quad \mathbf{b} = \begin{pmatrix} \mathbf{c} \\ \mathbf{d} \end{pmatrix} \quad (9)$$

where $[C_x | C_y | \mathbf{c}]$ and $[D_x | D_y | \mathbf{d}]$ are built from constraints (6) and (8), respectively. Notice that vector $\mathbf{c}$ is null as stated in (6). However, this constraint can be relaxed so as to introduce more flexibility into the layout. Figure 4 illustrates the resulting layout from solving the standard MIQP (5). Notice that the orthogonal order is accurately preserved while the boxes are thoroughly spread in the visualization window.
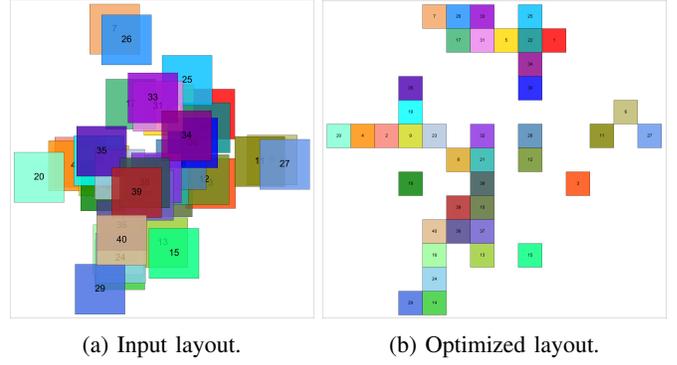

(a) Input layout.　　　　　(b) Optimized layout.

Fig. 4: Result of our method to rearrange boxes with overlaps.

IV. DEALING WITH MULTIPLE REQUIREMENTS FOR GEOMETRIC LAYOUTS ARRANGEMENT

In this section we describe a methodology for building layouts from geometric primitives which is able to deal with a wide range of requirements simultaneously. Relying on multidimensional projection, density-based adaptive grids, and mixed integer optimization, our approach is semantically aware, makes an efficient use of display area, and generates well structured grid-like layouts. Moreover, the formulation intrinsically impose a hierarchy on the data, enabling alternatives for the scalability issue.

The proposed optimization scheme arranges geometric entities (boxes) with varying sizes so as to avoid overlaps while preserving the neighborhood structure of the underlying data (semantics). The area of each geometric primitive is also included in the optimization process to ensure that the display area will be efficiently occupied. In fact, supported by the adaptive grid, our formulation is able to scale elements with different sizes using only one variable, thus rendering the optimization procedure as simple as possible.

The proposed methodology is composed by three main steps: *multidimensional projection*, *density-based adaptive grid generation*, and *layout optimization*.

In the first step, high-dimensional data is mapped to the visual space so as to preserve distance among data instances. In our implementation we use the Least Square Projection (LSP) method, which preserves distances nicely during the mapping process, thus enforcing that neighbor points in the visual space correspond to similar/close instances in the original space. The use of a distance preserving mapping to place instances in the visual space provides the neighborhood structure that must be "mimicked" by the final layout.

An adaptive grid is then generated from projected points. In contrast to typical adaptive grid generation schemes, our approach refines the grid in less dense (but not empty) regions of the visual space, thus placing larger grid cells in denser regions. The rationale is to allow users easily identify dense regions by visually recognizing large geometric objects in the layout. The number of refinement levels is a user defined parameter.

The adaptive grid produces a size varying tiling of the visual space. Moreover, grid cells inherit the semantic relation of the project instances, that is, neighbor grid cells tend to encompass similar instances. Although well structured, the cell arrangement resulting from the refinement is typically spread, making an inefficient use of display area. Therefore, in the third step of the proposed pipeline, cells are rearranged in the visual space to optimize the area usage. The optimization is formulated to account for object scale, overlapping, and grid-like arrangement while preserving neighborhood relationships and the aspect ratio among cells.

Let $\tilde{G} = \{g_1, g_2, ..., g_N\}$ be the set of non-empty cells in G, that is, $\tilde{G}$ comprises the cells of G with projected points in their interior. Each cell g_i is a square box described by the vector $g_i = (x_i, y_i, w_i) \in \mathbb{R}^3$, where (x_i, y_i) accounts for the center of the box and $w_i > 0$ is the edge length of g_i. The cells g_i should be rearranged and resized inside a $W \times H$ display area so as to make an efficient use of display area while preserving grid alignment and neighborhood structures. Additionally, $w_i = \alpha_i \delta$, where $\alpha_i = 1/2^k$, k corresponding to the level of refinement of g_i. Hence, δ is a parameter to be optimized from which the size of each cell is derived, as Figure 5 shows.

Rearrange cells g_i so as to generate layouts that optimize use of area while presenting a grid-like structure is an NP-hard problem present in different contexts. In order to get an approximate solution, we formulate the problem as a computationally tractable quadratic optimization problem. The optimization is formulated as in Equation (10) below:

$$\begin{aligned} \text{minimize} \quad & E(\mathbf{z}) = E_{comp}(\mathbf{z}) + E_{rezise}(\mathbf{z}), \\ \text{subject to} \quad & A\mathbf{z} \leq \mathbf{b}, \quad \mathbf{z} = [\mathbf{x} \ \mathbf{y} \ \mathbf{r} \ \delta]^\top, \\ & \mathbf{x} = (x_1, x_2, ..., x_N)^\top \in \mathbf{R}^N \\ & \mathbf{y} = (y_1, y_2, ..., y_N)^\top \in \mathbf{R}^N \\ & \mathbf{r} = (r_{12}, ..., r_{1N}, r_{23}, ..., r_{2N}, ..., r_{N-1N})^\top, r_{ij} \in \{0, 1\} \\ & \delta_0 \leq \delta \leq \min(W, H), \end{aligned}$$

(10)

where $\mathbf{z}$ is the sought solution; $\mathbf{y}$ and $\mathbf{x}$ correspond to the coordinates of the centroids of the cells; δ is the scaling factor; A and $\mathbf{b}$ hold the constraints imposed on the optimization problem. The unknowns r_{ij} are control variables used to properly avoid overlaps. The energy components $E_{comp}(\mathbf{z})$ and $E_{resize}(\mathbf{z})$ control the proximity between cells and the area increase, respectively. The former term accounts for overlaps and neighborhood preservation and the second term is designed to scale the box to fill up as much area as possible.

Figure 6 depicts qualitative results comparing our approach with overlap removal and visual board techniques, respectively. Notice that the proposed method gives rise to well structured layouts where neighborhoods (indicated by color map) are nicely preserved. Moreover, our approach makes a better use of display area, thus improving readability and content analysis.

One of our applications regards video visualization, as illustrated in Figure 7. We build a collection of 300 videos
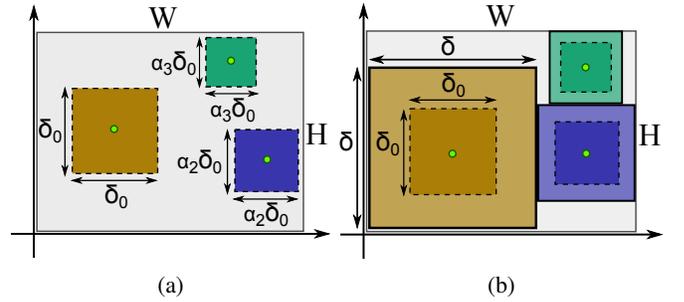

Fig. 5: (a) The input layout with δ_0 being the length of the coarser cell. The size of the remaining cells are defined according to the initial parameter δ_0 and their scale parameters α_i. (b) The result of the optimization when $\delta > \delta_0$.
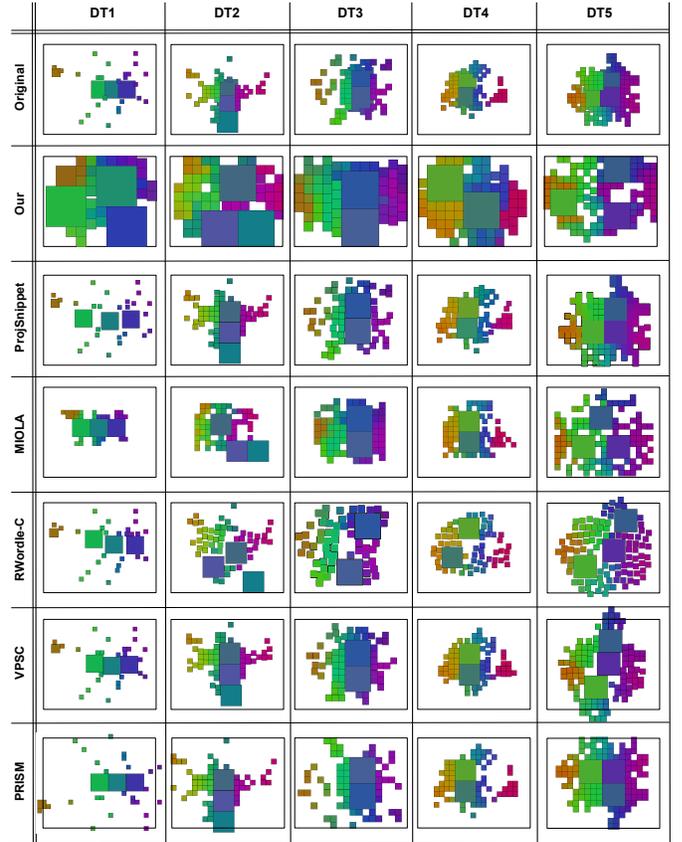

Fig. 6: Layouts produced by our approach, ProjSnippet, MI-OLA, RWordle-C, VPSC and PRISM in 5 datasets.

from Youtube querying six distinct topics, namely, linux, civil war, fifa world cup, hawk, guitar and information visualization. Textual information associated to each video is processed to generate a feature vector that represents the video in a high-dimensional space. The proposed methodology with three level grid refinement is employed to build a layout where larger cells are textured as word clouds while the cells in the lowest refinement level (smaller cells) are textured with a snippet build from a screenshot of a randomly chosen video contained in the cell as well as textual information containing title,

description and URL of the video. Word clouds are generated using keywords extracted from the text associated to the video.

Fig. 7: Visualizing 300 videos extracted from Youtube querying 6 different topics.

V. SEMANTICALLY AWARE DYNAMIC LAYOUTS

In this section we describe a novel semantic aware layout construction technique that allows users to freely tailor 2D arrangements according to their interest. The proposed formulation relies on interactive mechanisms enabled by multidimensional projection methods to enforce semantic relation in the layout. Moreover, the proposed approach is based on a simple energy function that can efficiently be minimized using well-known optimization libraries, thus avoiding intricate computational implementations while ensuring real-time layout updates. Similar to state-of-the-art techniques, our methodology is able to arrange geometric primitives in arbitrary visual domains, what renders it quite flexible and versatile.

The provided results show the effectiveness of our approach in building and organizing user tailored layouts. Semantic relation between entities derives from similarity metrics, therefore, the proposed methodology can be employed in different scenarios and applications.

The proposed mechanism to dynamically update layouts according to user intervention builds upon the methodology of ProjSnippet [1], which has been conceived to optimize layouts restricted to rectangular domains and with no interactive resource. Our approach, in contrast, enables interactive resources that allow users freely modify the layout according to their interest while still being able to build arrangements in arbitrary visual domains. Moreover, the proposed formulation combines the flexibility provided by control points used in multidimensional projection with an energy function tuned to enable interactive layout update as well as to enforce semantic relation among neighbor entities.

Figure 8 illustrates the use of our layout construction methodology in textual data analysis. More specifically, the

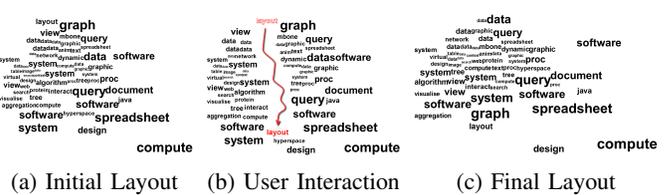(a) Initial Layout   (b) User Interaction   (c) Final Layout

Fig. 8: Word cloud with size varying primitives

layouts depicted in Figure 8 were generated by extracting bag-of-words from parts (abstract and title) of 60 visualization related papers published in the IEEE VisWeek Conference 2004. Therefore, each paper is represented by a term frequency vector in a high-dimensional space. After projecting data instances to the visual space, the most frequent word for each data instance is used to represent the paper in the visual space and to generate the word cloud. The size of each word in the layout reflects the relevance of its corresponding paper. Precisely, we build an histogram of words and papers with a large number of frequent words are considered more relevant. The result of projecting and optimizing the position of words is depicted in 8(a). Layouts generated during and after user intervention are depicted in Figures 8(b) and 8(c). More results can be found in http://youtu.be/cqua1VerGlo

VI. PUBLICATIONS, AWARDS AND HONORS

Publications (and Qualis)

[1] (*A1*), [2] (*B1*), [3] (*A1*), [4] (*B1*), [6] (*B1*), [7] (*A2*).

Awards and Honors

- Best paper in *Graphics and Visualization* at SIBGRAPI 2013
- Invited IEEE TVCG paper presented at IEEE Vis 2014
- Accepted proposal to Doctoral Colloquium at IEEE Vis 2015
- Invited IEEE TVCG paper presented at IEEE Vis 2016

REFERENCES

[1] E. Gomez-Nieto, F. S. Roman, P. Pagliosa, W. Casaca, E. S. Helou, M. C. F. de Oliveira, and L. G. Nonato, "Similarity preserving snippet-based visualization of web search results," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 3, pp. 457–470, 2014.

[2] E. Gomez-Nieto, W. Casaca, L. G. Nonato, and G. Taubin, "Mixed integer optimization for layout arrangement," in *Proceedings of the 26th Conference on Graphics, Patterns and Images*, ser. SIBGRAPI'13. Washington, DC, USA: IEEE Computer Society, 2013, pp. 115–122.

[3] E. Gomez-Nieto, W. Casaca, D. Motta, I. Hartmann, G. Taubin, and L. G. Nonato, "Dealing with multiple requirements in geometric arrangements," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 3, pp. 1223–1235, March 2016.

[4] E. Gomez-Nieto, D. Motta, and L. Nonato, "Semantically aware dynamic layouts," in *Graphics, Patterns and Images (SIBGRAPI), 2014 27th SIBGRAPI Conference on*, Aug 2014, pp. 220–226.

[5] 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," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 3, pp. 564 –575, 2008.

[6] E. Gomez-Nieto, W. Casaca, I. Hartmann, and L. G. Nonato, "Understanding large legal datasets through visual analytics," in *Proceedings...*, J. L. D. Comba and L. G. Nonao, Eds., Workshop on Visual Analytics, Information Visualization and Scientific Visualization, 6. (WVIS). Sociedade Brasileira de Computação, 2015.

[7] G. G. Zanabria, L. G. Nonato, and E. Gomez-Nieto, "istar (i*): An interactive star coordinates approach for high-dimensional data exploration," *Computers & Graphics*, vol. 60, pp. 107–118, 2016.