Generation of semantic layouts for interactive multidimensional data visualization

Erick Mauricio Gómez Nieto

Erick Mauricio Gómez Nieto

Generation of semantic layouts for interactive multidimensional data visualization

Doctoral dissertation submitted to the Instituto de Ciências Matemáticas e de Computação – ICMC-USP, in partial fulfillment of the requirements for the degree of the Doctorate Program in Computer Science and Computational Mathematics. *FINAL VERSION*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Prof. Dr. Luis Gustavo Nonato

USP – São Carlos
May 2017

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados fornecidos pelo(a) autor(a)

```
       Gómez Nieto, Erick Mauricio
G633g     Generation of semantic layouts for interactive
       multidimensional data visualization / Erick
       Mauricio Gómez Nieto; orientador Luis Gustavo
       Nonato. -- São Carlos, 2017.
          131 p.


          Tese (Doutorado - Programa de Pós-Graduação em
       Ciências de Computação e Matemática Computacional) --
        Instituto de Ciências Matemáticas e de Computação,
       Universidade de São Paulo, 2017.


          1. Overlap removal. 2. Similarity preserving. 3.
       Area optimization. 4. Semantic layout. 5. High-
       dimensional data. I. Nonato, Luis Gustavo, orient.
       II. Título.
```

Erick Mauricio Gómez Nieto

Geração de layouts semânticos para a visualização interativa de dados multidimensionais

Tese apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos para obtenção do título de Doutor em Ciências – Ciências de Computação e Matemática Computacional. *VERSÃO FINAL*

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientador: Prof. Dr. Luis Gustavo Nonato

USP – São Carlos
Maio de 2017

In Memoriam of Gomez Carbajal, Wilmer Pedro
the most creative, friendly, inspirational person and DAD
that I could have ever had.

ACKNOWLEDGEMENTS

First, I want to thank to God. Dude!, you really do things mysteriously but you're never wrong!.

I want to thank to the University of Sao Paulo in Brazil (USP) and its Institute of Mathematics and Computer Sciences (ICMC) for all these years of education, research and fun.

To my mentor and friend, Luis Gustavo Nonato, for challenge myself everyday during my PhD and MSc periods. Thanks for believe in me!.

To my family: Maria Nieto Nuñez (aka Mom), Katty Gomez Nieto (aka Cata), Doña Delia and Don Alberto Nieto and Patricia Nieto (aka Pato).

To my beloved fiance, Mili (aka bonis), thanks for all your support, love and patience along these years. There are no words enough to express my thankfulness and love.

To faculty professors of the ICMC-USP, specially for Maria Cristina Ferreira de Oliveira, Rosane Minghim, Afonso Paiva, Fernando Paulovich, Elias Helou and Adenilso Simão. Thanks for sharing with me your knowledge and experience, it always will be invaluable.

To my colleagues of Visual and Geometry Processing Group (VPGP) at ICMC-USP Wallace Casaca - Danilo Motta (meus bros), Douglas Cedrim (aka truta), Paola (Pao), Afonso Paiva (afopa), Luiz Fernando de Souza, Marcos Sandim, Filipe Nascimento and the visiting professors Gabriel Taubin, Stephen Payne, Paulo Pagliosa and Fabiano Petronetto.

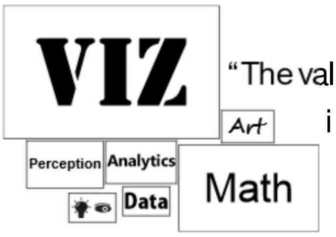
To my colleagues at IBM Research BRL during my internship there, specially for Rogerio de Abreu for giving me the opportunity to show my work, and Emilio, Matheus, Juliana, Leandro Fabiano, Débora, Carlos Cardonha, Andrea, Julio, Rodrigo, Miguel, Jorge and Vagner, I learned so much from you guys!.

To my bros Naim Mardini (aka Mostro) and Alex Cuadros-Vargas (aka Chefe).

To the music that always lit up this road, special thanks to Scott Weilland (RIP), Mr Mojo Risin (RIP), Eric Clapton, John Frusciante, Led Zepellin, Alice in chains, Gustavo Cerati, Steve Ray Vaughan, BB King, Eddie Vedder, Matt Bellamy, Nina Simone and for all those I forgot to mention.

An special thanks to Fundação de Amparo à Pesquisa do Estado de São Paulo - FAPESP (Process #2013/00191-0) and IBM Research for funding my research during my PhD period.

And to all people involved in my life during these 5 last years, I learned from all your knowledge, experiences and stories.



“The value of an <u>idea</u> lies
in the using of it.”
(Thomas A. Edison)

ABSTRACT

GOMEZ-NIETO, E.. **Generation of semantic layouts for interactive multidimensional data visualization**. 2017. 131 f. Doctoral dissertation (Doctorate Candidate Program in Computer Science and Computational Mathematics) – Instituto de Ciências Matemáticas e de Computação (ICMC/USP), São Carlos – SP.

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 ourapproaches.

Key-words: Overlap Removal, Similarity Preserving, Structured Layouts, Area Optimization, Semantic Layout, High-Dimensional Data.

RESUMO

GOMEZ-NIETO, E.. **Geração de layouts semânticos para a visualização interativa de dados multidimensionais**. 2017. 131 f. Doctoral dissertation (Doctorate Candidate Program in Computer Science and Computational Mathematics) – Instituto de Ciências Matemáticas e de Computação (ICMC/USP), São Carlos – SP.

Métodos de visualização fazem uso de representações gráficas interativas embutidas em uma área de exibição para exploração e análise de dados. Esses recursos visuais usam primitivas geométricas para representar dados ou compor representações mais sofisticadas que facilitem a extração visual de informações. Uma das tarefas mais desafiadoras é determinar um layout ótimo visando explorar suas capacidades para transmitir informação dentro de uma determinada visualização. Os algoritmos existentes para construir layouts a partir de primitivas geométricas são tipicamente projetados para lidar com requisitos como alinhamento ortogonal, remoção de sobreposição, área usada, organização hierárquica, atualização dinâmica entre outros. No entanto, a maioria das técnicas são capazes de lidar com apenas alguns desses requerimentos simultaneamente, prejudicando sua utilização e flexibilidade. Nesta tese, propomos um conjunto de abordagens para construir layouts a partir de primitivas geométricas que simultaneamente lidam com uma gama mais ampla de requerimentos. Baseando-se em projeções multidimensionais e formulações de otimização, os nossos métodos organizam objetos geométricos no espaço visual para gerar layouts bem estruturados que preservam a relação semântica entre objetos enquanto ainda fazem um uso eficiente da área de exibição. Um conjunto detalhado de comparações quantitativas com métodos existentes para a geração de layouts e aplicações em visualização de conjunto de dados de texto, imagem e vídeo comprova a eficácia das técnicas propostas.

Palavras-chave: Remoção de sobreposição, Preservação da similaridade, Layouts Estruturados, Otimização de Área, Layout Semântico, Dados em Alta Dimensão.

LIST OF FIGURES

LIST OF CHARTS

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

AU	Area Usage
CMDS	Constrained Multidimensional Scaling
D	Display area (in figures)
Du	Dunn Index
ED	Euclidean Distance
HSV	Human Visual System
InfoVis	Information Visualization
L	Orthogonal Alignment
LS	Layout Similarity
MP	Multidimensional Projection
NP	Neighborhood preservation
OO	Orthogonal Ordering
SI	Size increase
SOM	Self-Organizing Maps
SP	Shape preservation
TP	Topology Preservation
VA	Visual Analytics

CONTENTS

CHAPTER

1

INTRODUCTION

1.1 Motivation

Arranging geometric primitives to generate meaningful layouts is a major task in visualization, which inherently appears in important applications such as word cloud construction (KOH et al., 2010; PAULOVICH et al., 2012), small-multiples arrangements (BERTIN, 2007; TUFTE, 1990; ELZEN; WIJK, 2013), and visual boards (PINHO; OLIVEIRA; LOPES, 2010; STRONG; GONG, 2011; STRONG; GONG, 2014). The difficulty in building layouts made up a large number 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 (FEKETE; PLAISANT, 2002) 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 such as RWordles (STROBELT et al., 2012) and ProjCloud (PAULOVICH et al., 2012) 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 address this challenging problem by proposing new techniques for building layouts. We denominate by 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.

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

1.2 Thesis statement

Organizing multidimensional data into semantic layouts improves information discovery and data analysis tasks. These tasks demand the optimization of multiple conflicting requirements, such as semantic preservation, overlap removal, area usage optimization, object scaling, orthogonal alignment and dynamic update. Properly address those issues allow for an easier and effective exploration of complex multidimensional data.

This document presents a compilation of different techniques for interactive 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 proposed methods rely on novel optimization formulations for simultaneously dealing with requirements that were identified as the most relevant during our study. The last method presented in this dissertation focuses on an application that demands semantic preserving layout updates during analyst's interactive data exploration.

In summary, this dissertation gathers techniques and results developed and published during the doctorate term. Specifically, it puts together the results from the following publications:

- "Similarity Preserving Snippet-Based Visualization of Web Search Results" (GOMEZ-

NIETO et al., 2014)
The contributions of this work are:

  - We propose a method, which we called ProjSnippet, to display the results of textual queries integrating textual snippets and multidimensional projection into a simple and intuitive visualization.
  - The ProjSnippet relies on a novel overlap removal energy functional that considers both the neighborhood relations between snippets and their overlapping in the visual space.

- “ Mixed Integer Optimization for Layout Arrangement” (GOMEZ-NIETO et al., 2013) The contributions of this work are:

  - We propose a mathematical formulation, which we call MIOLA (Mixed Integer Optimization for Layout Arrangement), for the problem of arranging rectangular boxes in the visual space so as to place similar entities close to each other while avoiding overlaps.
  - MIOLA combines flexibility and capability in generating different layouts while still solving the box-overlap removal problem.
  - A video visualization application using MIOLA that allows for handling and exploring video data sets.

- “ Dealing with Multiple Requirements in Geometric Arrangements” (GOMEZ-NIETO et al., 2016) The contributions of this work are:

  - We propose a methodology to generate layouts made up of geometric primitives which deals with multiple requirements such as grid-like structure, semantic relation among objects, display area usage, and overlap removal. The methodology naturally imposes an hierarchy on the visual representation in order to handle a large number of data instances.
  - We propose an optimization scheme that relies on a reduced number of unknowns to generate well structured layouts while making a good use of display area.
  - We present a comprehensive set of comparisons against existing methods and several practical applications to confirm the effectiveness and usefulness of this methodology.

- “ Semantically Aware Dynamic Layouts” (GOMEZ-NIETO; MOTTA; NONATO, 2014) The contributions of this work are:

  - We propose one fully interactive technique to arrange geometric primitives in two-dimensional layouts.

- A mechanism to enforce semantic relationships among entities of the layout.
- A combination of interpolation and optimization mechanism to enable a pleasant and dynamic layout update during and after user intervention.

1.3 Outline

This thesis is organized in the three main parts: The first part (chapters 2 and 3) covers specific concepts on semantic layout generation and relevant literature to contextualize the problem addressed. The second part (chapters 4 to 7) presents in detail the proposed methods. Finally, the third part (chapter 8) present the conclusions and future work.

In summary the subjects covered by each chapter are:

- Chapter 2: On Semantic Layouts Requirements presents an study of general concepts to contextualize the problem of semantic layout generation. It is divided in three main sections: requirements, operations and quantitative metrics.
- Chapter 3: Review of the Literature describes a set of relevant methods to semantic layout generation, organized into six different categories: overlap removal, board-based, space-filling, alignment, hierarchy-based and dynamic update methods.
- Chapter 4: ProjSnippet, is a visualization technique to display the results of web queries through rectangular element layouts. It combines the neighborhood preservation capability of multidimensional projections with the familiar snippet-based representation.
- Chapter 5: MIOLA, is a mechanism to arrange rectangular boxes in a two-dimensional layout which keeps similar objects close while preventing overlaps. In contrast to heuristic techniques, our approach relies on mixed integer quadratic programming, resulting in well structured arrangements which can easily be tuned to take different forms.
- Chapter 6: Dealing with Multiple Requirements presents a novel methodology for building layouts from geometric primitives that concurrently addresses a wider range of requirements. Relying on multidimensional projection and mixed integer optimization, our approach arranges 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.
- Chapter 7: Semantically aware Dynamic Layouts describe a novel interactive semantic aware layout construction technique that relies on a simple mathematical and computational formulation to allow users to freely interact with the layout. This approach is supported by interactive multidimensional projection methods, which enforces similar instances to be placed close to each other during layout updates.

- Chapter 8: Conclusion and Future Work summarizes our general conclusions, discusses limitations found during our study and describes some future works that can be conducted to further develop this research topic.

CHAPTER

2

ON SEMANTIC LAYOUTS REQUIREMENTS

2.1 Introduction

This chapter provides basic concepts as to semantic layouts generation. First, we list a collection of requirements that should be cared for during semantic arrangement of geometric entities. The second section describes a set of operations that deals with such requirements. Finally, the last section presents a set of suitable metrics compiled from literature to assess the performance of those operators.

2.2 Semantic layout arrangement requirements

In this section, we present a detailed list of requirements for semantic layouts generation that we have identified as the most relevant ones during our studies. In order to better present the achievements of our studies, this section divided into seven subsections, namely, similarity preserving, area usage, layout structure, content, perceptive features, interactivity and adaptability, which are described below.

2.2.1 Similarity preserving

Similarity is one of the most used resources for data visualization. It allows the analyst to discover patterns, trends or relationships in data through grouping of semantically related instances, i.e. data with similar values in its representative feature vector. In general, similarity is geometrically represented by spatial proximity according to a distance measure, e.g. Euclidean, Manhattan or Mahalanobis distances.

The following methods use similarity to map and organize high-dimensional data into 2D visual space. These methods are highly useful in the context of our work, so they will be presented here.

2.2.1.1 Multidimensional projection

Multidimensional Projection (MP) relies on dimensionality reduction theory to embed multidimensional data into a visual space. Thus, it allows the visual exploration of different types of data, such as, document collections (PAGLIOSA et al., 2013), image datasets (JOIA et al., 2012) or even scientific data such as vector fields (DANIELS et al., 2010). Formally, multidimensional projection methods can be defined as follows:

Definição 1. Multidimensional Projection (TEJADA; MINGHIM; NONATO, 2003) Let X be a set of points in $\Re^n$ and $\partial : \Re^n \times \Re^n \rightarrow \Re$ be a criterion of proximity between points in $\Re^n$. A multidimensional projection is a mapping f that takes X to a set of points P in $\Re^v$, $v = \{1, 2, 3\}$ such that if $f : X \rightarrow P$ is a bijective relation and $d : \Re^v \times \Re^v \rightarrow \Re$ is a proximity criterion in $\Re^v$ then $\partial(x_i, x_j) - d(f(x_i), f(x_j))$ is as close to zero as possible, $\forall x_i, x_j \in X$.

There exist different taxonomies for classifying multidimensional projection techniques. For instance, Paulovich, Silva e Nonato (2010) classified the MP methods into three main categories, namely, spectral decomposition, nonlinear optimization, and force-based schemes. Another classification was proposed by Joia et al. (2011), grouping them into local and global methods according to the information used to perform the projection. Global methods use only a single transformation, while local techniques make use of neighborhood information of each instance and the location of a subset of samples a priori positioned in the visual space to define a family of mappings. Recently, Lehmann e Theisel (2016) classify MP methods based on the type of projection used to map data, i.e. projective projections, affine projections and distance-based projections.

It is worth to mention that most traditional methods in literature, such as FastMap (FALOUTSOS; LIN, 1995), Multidimensional Scaling (MDS) (TORGERSON, 1952) and Kernel Principal Component Analysis (Kernel PCA) (SCHöLKOPF; SMOLA; MüLLER, 1998), established the foundations for most up-to-date techniques. In fact, recent developments have extended MP methods capabilities with remarkable success, for instance, reducing the number of distance information to be computed (LSP (PAULOVICH et al., 2008), PLMP (PAULOVICH; SILVA; NONATO, 2010)), interactive manipulation of the projection matrix (LAMP (JOIA et al., 2011), iPCA (JEONG et al., 2009)), create inverse projection from 2D space back to a high-dimensional space (iLAMP (AMORIM et al., 2012)) or control the influence of parameters in the projection process (Probing Projections (STAHNKE et al., 2016)).

In general, the approaches presented in this dissertation use LSP and LAMP as projection mechanism due to their good accuracy in terms of distance preservation and low computational cost.

2.2.1.2 Self-organizing maps

Self-Organizing Maps (SOM) is an artificial neural network based method that makes use of data compression process, known as vector quantisation, for reducing dimensionality of vectors. The traditional Kohonen's (KOHONEN, 1982) approach is typically employed in this context (GASSEN et al., 2015). In terms of layout, it provides an optimal use of display area since the layout is a priori defined.

More sophisticated methods were proposed extending the Kohonen approach. For instance, Lagus, Kaski e Kohonen (2004) use the same principle of compression for summarizing and mining large document text collections, named WEBSOM. Recently Strong e Gong (2014) proposed Self-Sorting Maps (SSM), an greedy-based coarse-to-fine algorithm to compute near optimal data placements of multimedia data.

2.2.2 Area usage

Since display devices are limited by their fixed physical space, a proper management of area is a fundamental requirement for all disciplines of Visual Computing[1], as InfoVis and VA. Most visualizations are displayed into a workspace (also known as visual space) that will represents the whole space available for displaying visual resources, even if multiple devices are used for the same application, as in Video Wall's case.

In visualization, there exist different methods to exploit the visual space. Some methods are devoted to split recursively the whole space into non-overlapping regions allocating data to one of these regions, e.g. Treemaps. Other methods build several regions to represent data already arranged regardless important considerations as to overlap or area usage – since the sum of areas of all regions is not (usually) equal to visual space area – but preserving the original arrangement of the objects represented.

However, the empty space plays an important role for conveying similarity. For instance, grouping is clearly defined by the presence of a space that isolates a group for an easy and intuitive visual discovery. Thus, finding a balance between area usage maximization and any other perceptive feature for improving visual exploration is a complex task. This is one of the main issues addressed in this work.

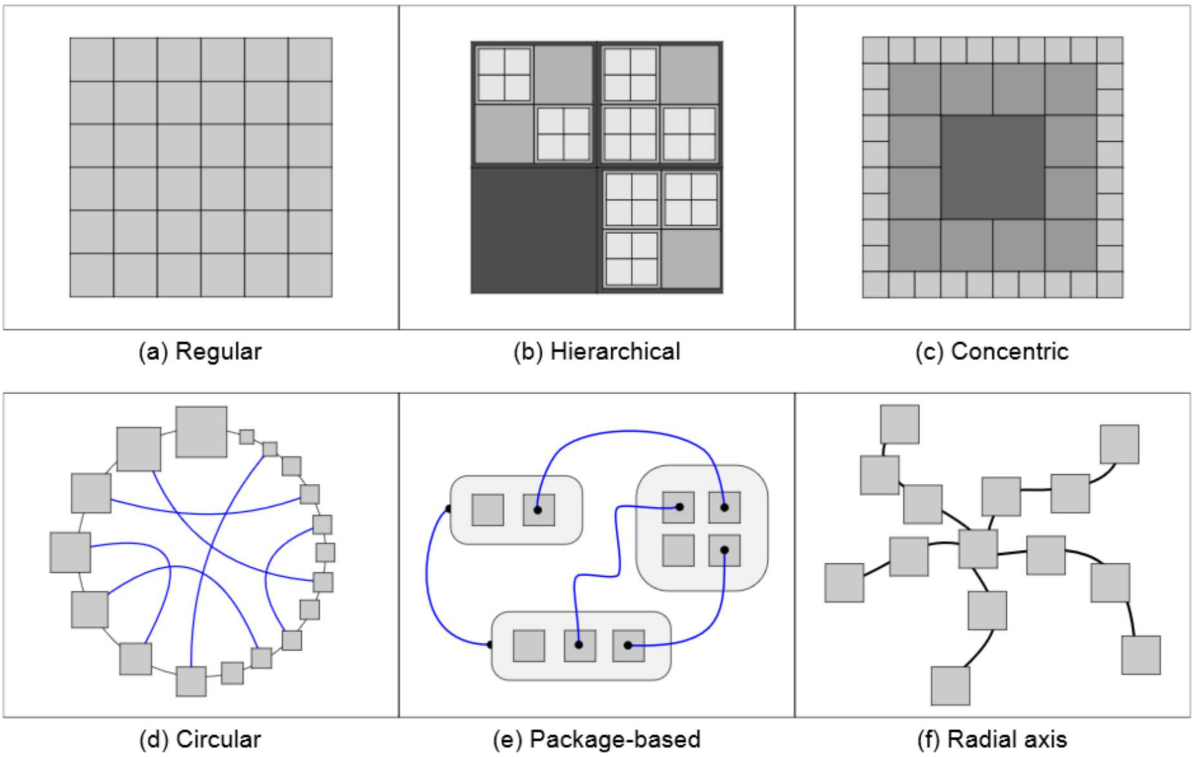
2.2.3 Layout structure

Generally, a layout structure can be seen as the manner that a set of objects are arranged or distributed on a limited space. This distribution varies according to what will be conveyed. As Figure 1 illustrates, different types of layouts transmit different ideas. For instance, a regular layout – without differences in size or any other perceptive feature – conveys an equal-weighted

[1] Generic term to group all computer science disciplines handling with images and graphics, as computer graphics, image processing, visualization, visual analytics, computer vision and virtual/augmented reality

focus of attention for all instances, in contrast to a concentric layout which centers the major importance on a specific entity, as shown in Figures 1(a) and 1(c) respectively. Figure 1(b) illustrates a hierarchical layout where each level is represented by a different color, i.e. as the higher the level is, the darker the corresponding color. Circular layouts can be used to represent list of objects following an specific ordering, with the purpose of use inner space to represent relationships among them, as shown in Figure 1(d). Package based layouts allow to improve readability and understandability of highly linked representation, e.g. large graphs with a high number of connected nodes, as Figure 1(e) shows. To this end, optimization methods are used to formulate automatic mechanism of compression. For a up-to-date review see Tim Dwyer's recent works (YOGHOURDJIAN et al., 2016), (DWYER et al., 2014) and (DWYER et al., 2013). Radial axis layouts group subsets of objects in axes around a central object according to a established metric (e.g. similarity, centrality or class), as shown in Figure 1(f).

Figure 1 – Six different types of layouts to arrange geometric entities. Linked nodes are joined by a blue edge.



(a) Regular

(b) Hierarchical

(c) Concentric

(d) Circular

(e) Package-based

(f) Radial axis

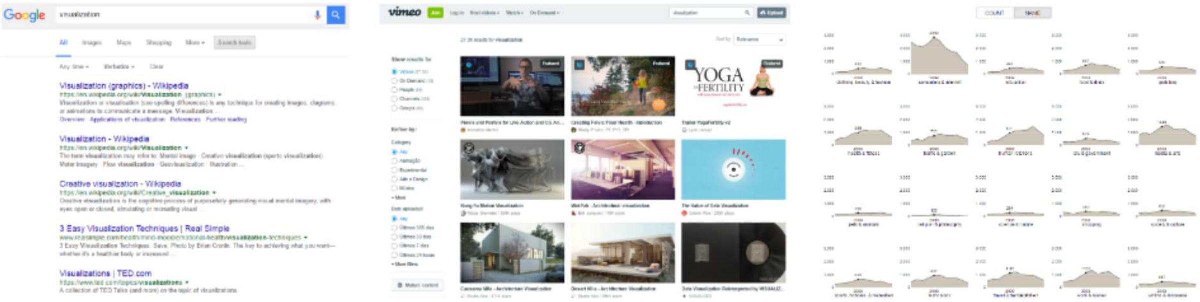
Source: Elaborated by the author.

2.2.4 Content

One of the main purposes when using geometric entities is to provide suitable alternatives to display information associated to instances. Many visualization techniques try to generate compact representations that summarize the whole content of each instance for an easy and fast inspection of data. Web search engines, for instance, use textual snippets that depict a title, a brief description and URL of websites. Some other websites focused on video streaming (e.g.

Youtube, Vimeo) add an image thumbnail to illustrate the video content, also known as visual snippets. Moreover, some visual methods embed abstract representation into each geometric entity area as barcharts, linecharts, scatterplots, wordclouds or sophisticated methods as Radial visualizations (e.g. Star coordinates (KANDOGAN, 2001), RadViz (HOFFMAN et al., 1997)) or Parallel Coordinates (INSELBERG, 1998), as shown in Figure 2.

Figure 2 – Different types of content into geometric entities: (a) textual snippets extracted from Google (<http://www.google.com>), (b) visual snippets extracted from Vimeo (<http://www.vimeo.com>), and (c) small multiples using line charts extracted from Flowingdata.com (<http://projects.flowingdata.com/tut/linked_small_multiples_demo/>)



Visualization (graphics) - Wikipedia

Visualization - Wikipedia

Creative visualization - Wikipedia

3 Easy Visualization Techniques | Real Simple

Visualizations | TED.com

(a) Textual snippets

(b) Visual snippets

(c) Small multiples

Source: Elaborated by the author.

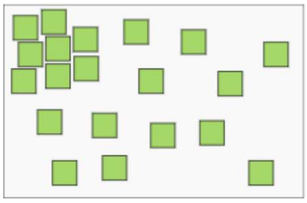
2.2.5 Perceptive features

2.2.5.1 Preattentive processing

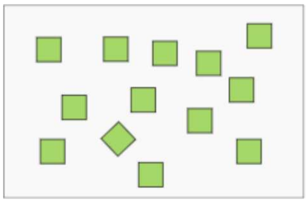
Currently, the term “preattentive” is commonly used for conveying an intuitive notion of some visual properties that can be accurately and rapidly detected by Human Visual System (HSV) and that precede the focused attention (HEALEY; ENNS, 2012). In general, actions performed in less than 200-300 milliseconds (msecs.) can be considered preattentive since eye movements take around 200 msecs. to initiate and focus on any particular location.

Below, we described a subset of visual features considered as preattentive related to layout arrangement of geometric entities. For a more complete review of preattentive elements in visualization see (HEALEY; ENNS, 2012) and (WARE, 2004).

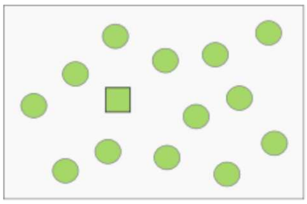
Hue is the simplest example of a preattentive property. The orange square, illustrated on the inline image, can be easily identified among the rest of green instances. The more different the target's hue is, faster the detection process. In this context, an interesting study of color applied to cartographic maps is presented in ColorBrewer (HARROWER; BREWER, 2003).



Density is a visual property that naturally conveys grouping. Most visualizations that aim to represent similarity make use of density to generate visual heaps according to an specific measure, as illustrate the inline figure. Additionally, some well-known clustering algorithms exploit spatial density among data to determine groups automatically, as k-Means (LLOYD, 1982) or DBScan (ESTER et al., 1996).



Orientation is a very used element on perceptive textures generation. Julész et al. presented to the psycophysics community a model based on Textons – elemental microstructures in natural images, considered as the elemental unit on human visual perception. For more information about this topic see (JULESZ, 1981) and (JULESZ, 1984)



Shape is a visual element that can be used to emphasize dissimilarity among data types, as illustrate the inline figure. Besides it is an important property related to different requirements on layout arrangement, e.g. area usage is strongly influenced by the geometric shape used to display instances since design area can be totally exploited by polygons as squares or triangles, in contrast to circle shapes. A suitable set of shapes should follow the same guidelines as the used in tessellation process in computer graphics area (SEYMOUR; BRITTON, 1989).

Size is, undoubtedly, one of most perceptive features on visualization. Frequently, size is an element used to visually convey values of a quantitative feature limited by a range, such that, instances with large associate values should make use of larger area in the visual space, as illustrate the inline figure.

Direction of motion concerns the vector direction of displacement, for example as consequence of a specific event (e.g. click, drag or any other mouse event). It can be used to convey similarity in a dynamic update process, for instance, when an specific instance is dragged along a semantic map.

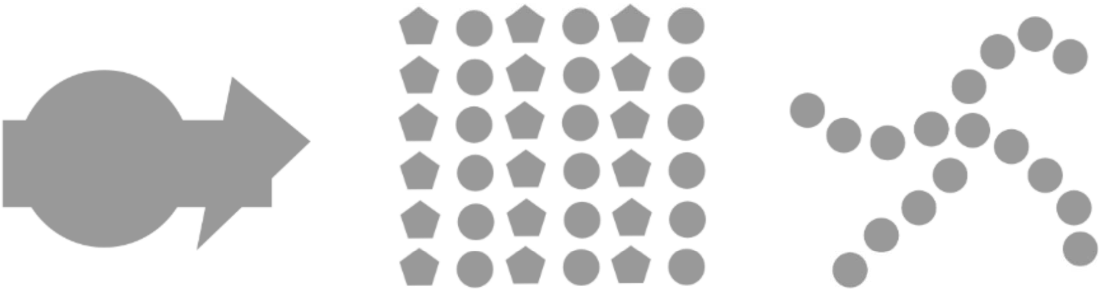
Velocity of Motion is another feature of dynamic update representations. The variation on displacement speed is visually easy to identify and it can be attributed as a feature as well. Thus, during a process updating the position of instances, this feature can perceptively convey the notion of flow quick instances are moving.

2.2.5.2 Gestalt principles

They are also called "Psychology of Form", it is basically made up of the seven[2] Fundamentals of Gestalt: continuity, symmetry segregation, prägnanz, closure, similarity and proximity. It assumes that the visual perception is a holistic process[3]. This implies that the underlying structure of a visual system is more important than the elements that compose it. This affirmation is typically summarized as " The whole is more than the sum of its parts" .

Typically, closure, similarity and continuity are the three most relevant fundamentals of Gestalt, illustrated in Figure 3. Closure says that human visual perception tries to first look at simpler objects when analyzing a complex arrangements of individuals, similarity state that tend to relate objects that are similar better than objects that are dissimilar, and continuity states that learners "tend to continue shapes beyond their ending points".

Figure 3 – Illustration of three fundamental Gestalt laws.



(a) Closure law (b) Similarity law (c) Continuity law

Source: Adapted from Keim e Zhang (2011).

2.2.6 Adaptability

Many technologies allow us to display visual representations into different devices (e.g. Wall displays, PCs, tablets, smartphones), but not all of them provides a suitable environment for interaction when we port these visual resources from one to another device. For instance, visualizing large and dense graphs turns into an intricate and unpleasant experience on small devices. In the specific case of layouts composed by 2D geometrical entities, adaptability is

[2] This number can be vary depending on the source

[3] It is based on the premise that properties of any system (the human for example) can be determined by analysis or treatment of its parts

strongly impacted by the number of elements displayed. In general, interactive operations for data exploration in small display devices are not so pleasant as in large screen devices as they provide more space to freely organize elements. However, this resctriction is considered for generating adaptable and ultra-compact layouts, as proposed by Yoghourdjian et al. (2016).

2.2.7 Interactivity

An interaction technique is essentially a fusion of input and output involving hardware and software elements and providing a way for users accomplish a task (TUCKER, 1996). In InfoVis, it plays an important role allowing users to explore and understand large amounts of information at once.

In the specific case of semantic layouts arrangement, interactive methods perform a wide set of operations. After semantic relationships being established, the analyst explores the map in order to identify new patterns and understand the reasons for those relationships, so this process can involve a repositioning of the instances, or some theory of set operations (e.g. union, separation, intersection). For this, the main challenge is to preserve semantic during/after updating process.

2.3 Operations

In this section, we describe a set of suitable operations for dealing with the layout generation requirements mentioned above. We have categorized them into six types according to the purpose to be accomplished, namely, overlap removal, space-filling, orthogonal ordering and alignment, grouping/clustering, hierarchy generation and dynamic update. We describe these categories in the following.

2.3.1 Overlap removal

Depending of visual metaphor used, overlap is the main difficulty for data exploration[4]. There exist different factors causing occlusion between elements in visualization techniques, but in general, it is due to an insufficient area of the visual space or a non-optimal overlap removal method for the graphical elements arrangement.

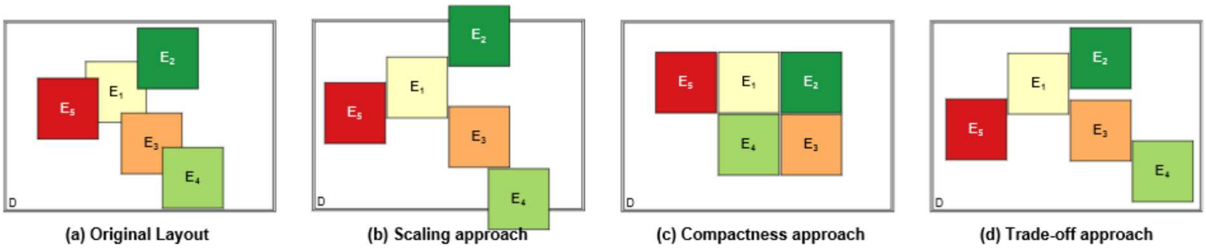
For our purposes of semantic layouts generation, overlap removal has a crucial importance. The simplest approach is to shrink objects until overlap disappears, as Figure 4(b) shows. Albeit this strategy preserves the original layout shape, it is inefficient in terms of compactness and area usage. The overlap removal problem has also been modeled as a packing problem where geometric entities are arranged such that they occupy a limited area, disregarding neighborhood

[4] In some visualization techniques, overlapping is used to convey grouping of non-classified data, for instance Parallel Coordinates (INSELBERG, 1998), Star Coordinates (KANDOGAN, 2001) or Graph Splatting (LIERE; LEEUW, 2003).

relations as Figure 4(c) illustrates. Generally, these methods use numerical optimization to find an optimal configuration for the layout.

As can be noticed, the two strategies mentioned above maximize only one requirement disregarding any other. Therefore, an overlap removal technique that deals simultaneously with both neighborhood preservation and area usage optimization arises as a suitable solution for generating compact and semantic-aware layouts, as Figure 4(d) illustrates. However, the formulation of this solution is not a trivial task since it becomes more complex as more requirements are simultaneously considered, as will be evidenced in the following chapters.

Figure 4 – Three approaches to solve the overlap removal preserving semantic problem.



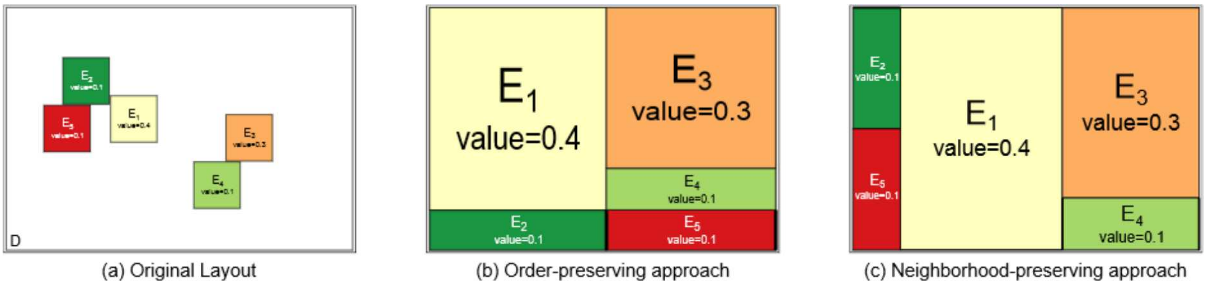
(a) Original Layout (b) Scaling approach (c) Compactness approach (d) Trade-off approach

Source: Elaborated by the author.

2.3.2 Space-filling

Essentially, this requirement accounts for maximal usage of area. As commented previously in Section 2.2.2, many techniques consider a value (usually known as weight) to set up the contribution on the area splitting process, as can be seen in Figure 5 where two different approaches which differ on the order they allocate geometry entities are shown. Figure 5(b) shows an arrangement based on area size – larger elements placed first – similar to traditional treemaps strategies. Figure 5(c) illustrates a balanced arrangement that, simultaneously, tries to preserve neighborhood relation from the original layout, i.e. adjacent entities in the original layout should be adjacent in the final layout.

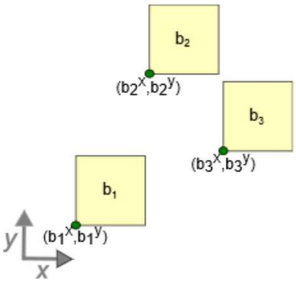
Figure 5 – Two different approaches to solve the space-filling problem.



(a) Original Layout (b) Order-preserving approach (c) Neighborhood-preserving approach

Source: Elaborated by the author.

2.3.3 Orthogonal ordering and alignment



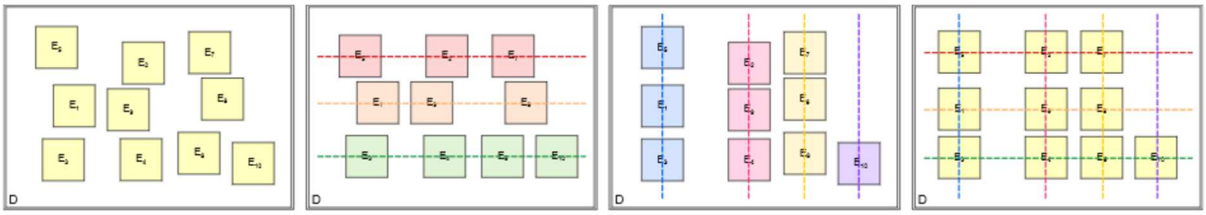
These operations aim to enforce orthogonality among geometric entities. Suppose a layout containing three boxes ($B = \{b_1, b_2, b_3\}$) in 2D space where the bottom-left corner of each box is denoted as b_1^x, b_2^x, b_3^x for x-coordinates and b_1^y, b_2^y, b_3^y for y-coordinates (as illustrated in the inline figure). The relative order of the objects is preserved if the following relation is ensured:

$$\begin{aligned} b_1^x < b_2^x < b_3^x \quad (x-\text{ orthogonal order}) \\ b_1^y < b_3^y < b_2^y \quad (y-\text{ orthogonal order}) \end{aligned} \tag{2.1}$$

Therefore, moving the boxes while ensuring inequality above, can generate layouts that preserves the initial neighborhood structures.

Another useful operation is the alignment of geometric primitives. It can be performed by setting the same value in y-coordinates (for horizontal alignment or H-Align) and x-coordinates (for vertical alignment or V-Align) for a subset of elements. Figures 6(b-c) illustrate H-Align and V-Align performed on three subsets of instances. Additionally, Figure 6(d) shows the resulting layout after perform sequentially both operations based on horizontal and vertical nearest neighbor for each element. This last operation is also called layout regularization.

Figure 6 – Three different set of constraints for orthogonal alignment: (a) Original layout, (b) H-Align only, (c) V-Align only, and (d) Both alignments sequentially performed based on horizontal and vertical nearest neighbor for each element



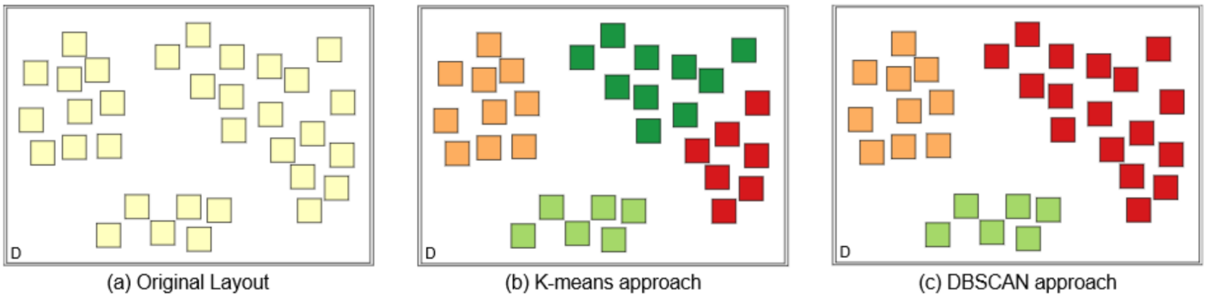
(a) Original Layout (c) H-Align only (b) V-Align only (d) H-Align then V-Align

Source: Elaborated by the author.

2.3.4 Grouping/Clustering

Analogously to point-based representations, grouping can be applied to layouts composed of geometric entities. Well-known clustering methods can be performed considering a common spatial position for all entities, e.g. centroid or minimal point. Figure 7 illustrates the use of the k-means (LLOYD, 1982) and DBScan (ESTER et al., 1996) algorithms for clustering instances in a geometric layout.

Figure 7 – Two different approaches for data clustering.



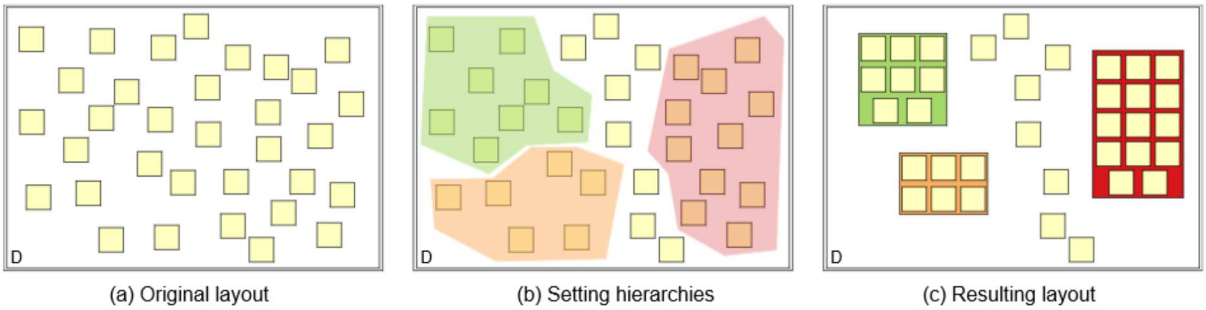
(a) Original Layout (b) K-means approach (c) DBSCAN approach

Source: Elaborated by the author.

2.3.5 Hierarchy generation

When the number of geometric entities to be displayed is large, the available area can be insufficient to provide a suitable visualization. A straightforward solution to this issue is to use hierarchical representations. Hierarchies can be obtained through user interaction – based on his/her preferences – or by grouping mechanisms as detailed in Section 2.3.4 and illustrated in Figure 8.

Figure 8 – Setting hierarchies by grouping of elements.



(a) Original layout (b) Setting hierarchies (c) Resulting layout

Source: Elaborated by the author.

2.3.6 Dynamic update

This operation concerns about the behavior of entities after any event or user interaction. To accomplish this task, different approaches have been proposed, as the use of incremental models for organize data such that each time a new item will be added (PINHO; OLIVEIRA; LOPES, 2010) or similarity-preserving mechanisms for reorganizing fixed-size sets of elements (REINERT; RITSCHEL; SEIDEL, 2013). In semantic layout case, updating the layout dynamically is accomplished by generating maps which preserves semantic relations during the entire exploration process.

2.4 Quantitative metrics

In this section, we present a set of ten quantitative metrics to measure the quality of final layout after performing any of the operations described in Section 2.3 on the initial (or original) layout, in terms of the requirements presented in Section 2.2.

2.4.1 Euclidean distance (ED)

Denoting the bottom-left corner of each box before and after overlap removal by x_i^o and x_i, the Euclidean distance metric is defined as

$$E = \frac{1}{n}\sum_{i=1}^{n} d(x_i^o, x_i) \tag{2.2}$$

where n is the number of boxes and d is the 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.

2.4.2 Layout similarity (LS)

This metric attempts to quantify how much neighborhood structures are affected by the overlap removal mechanism and it is derived from the Frobenius metric. The idea is to measure how much the length of Delaunay edges, computed from the original layout, changes after overlap removal. In mathematical terms, letting l_{ij}^o and l_{ij} denote the lengths of the Delaunay edges before and after overlap removal, the layout similarity is given by

$$\sigma = \frac{\sqrt{\sum (r_{ij} - \bar{r})^2 \;/ m}}{\bar{r}}, \quad \bar{r} = \frac{\sum r_{ij}}{m} \tag{2.3}$$

where $r_{ij} = l_{ij}/l_{ij}^o$ and m is the number of edges in the Delaunay triangulation.

2.4.3 Size increase (SI)

Given the convex hulls C^o and C of the original and modified layouts, the size increase is measured as:

$$S = \frac{\text{area}(C)}{\text{area}(C^o)} \tag{2.4}$$

determining the relative changes in size as well as the compactness of the representation.

2.4.4 Shape preservation (SP)

Strobelt et al. (2012) proposed a measure based on shape variation between the original (C^o) and modified (C) convex hulls.

The lengths len^o_α and len_α are computed for angular orientations from 0°- 350° in $\alpha = 10°$ steps from the centers of masses to intersections with each convex hull, respectively. The differences d_α describe the shape increase towards the sampled directions.

$$d_s = \frac{len_s}{len^o_s} \tag{2.5}$$

Then, standard deviation of all differences values is computed:

$$SP = \sqrt{\frac{1}{36}\sum_{s=0}^{35}(d_{s\times\alpha} - mean(d))} \tag{2.6}$$

Intuitively, it can be inferred that if a shape is equally expanded on the $s \times \alpha$ directions the standard deviation should be small, and if the shape is strongly deformed it should be high.

2.4.5 Neighborhood preservation (NP)

Proposed by Paulovich e Minghim (2008), this metric was originally used to assess multidimensional projection techniques performance. It computes the average percentage of preservation of the k-nearest neighbors of each box in the final layout. It can be seen as a derivation of the Topology Preservation measure, detailed below.

2.4.6 Topology preservation (TP)

König (2000) compares the rank order of neighbors between the original and visual space using topology information. Denoting by $NN_{ji}(i \in [1,k], j \in [1,m])$ and $nn_{ji}(i \in [1,k], j \in [1,m])$ the k nearest neighbors of instance j in the original and visual space respectively, the rank order of each instance j is assessed by the following credit assignment:

$$r_j = \begin{cases} 3, \text{ if } NN_{ji} = nn_{ji} \\ 2, \text{ if } NN_{ji} = nn_{jl}, l \in [1,k], i \neq l \\ 1, \text{ if } NN_{ji} = nn_{jt}, t \in [k,s], k < s \\ 0, \text{ otherwise} \end{cases} \tag{2.7}$$

where s is a fixed number. Typically $k = 4$ and $s = 10$. The global topology preservation is:

$$r = \frac{1}{3mk}\sum_{j=1}^{m} r_j \tag{2.8}$$

The value $r \in [0,1]$ where $r = 1$ is the perfect topology preservation.

2.4.7 Orthogonal ordering (OO)

Measures the number of changes in both horizontal and vertical ordering of a given set of geometric entities (MISUE et al., 1995). More precisely, positions are sorted in increasing horizontal and vertical order and the number of inversions in the lists provides the quality measure. In mathematical terms:

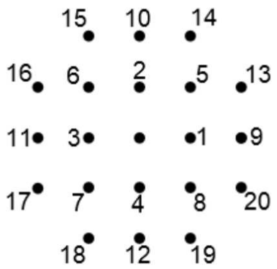
$$OO = \sum_{s \in \{x,y\}} \sum_{i<j} inv_s^{(t)}(i,j), \tag{2.9}$$

$$inv_x^{(t)}(i,j) = \begin{cases} 1, & \text{if}(x_i^{(t)} - x_j^{(t)})(x_i^{(t-1)} - x_j^{(t-1)}) < 0 \\ 0, & \text{otherwise} \end{cases} \tag{2.10}$$

where i, j are indices of the sorted lists before $(t-1)$ and after (t) layout arrangement. $inv_y^{(t)}(i,j)$ is defined analogously to $inv_x^{(t)}(i,j)$. Small values indicate better behavior.

2.4.8 Orthogonal alignment (L)

Let $\{z_1, z_2, \ldots, z_r\}$ be points in $Z \times Z$ discretizing the interior of a disk, as illustrated in the inline figure. Moreover, let b_i be the center of a square in the layout and b_{ij}, $j = 1, \ldots, k$ be the center of the squares in the k-nearest neighbor of b_i. The orthogonal alignment metric is given by:



$$L = \frac{1}{nk} \sum_{i=1}^{n} \sum_{b_{ij}} \max_{s} \left(\frac{< z_s, b_{ij} - b_i >}{\|z_s\| \|b_{ij} - b_i\|} \right), \tag{2.11}$$

where n is the number of geometric objects in the layout and $< \cdot, \cdot >$ accounts for the dot product. This metric gauges how much a layout deviates from a regular grid, the closer to 1 the better.

2.4.9 Area usage (AU)

Let r be the longer diagonal of the bounding box of the layout produced by a given method and A_i the area of the i-th box in the layout. The area usage metric is defined similarly as (SAMET, 2005):

$$AU = \frac{2\pi r^2}{\sum_i A_i}. \tag{2.12}$$

This metric gauges layout compactness and values close to $\pi/2$ are better.

2.4.10 Dunn Index (Du)

This metric (XU; WUNSCH, 2009) identifies clusters that are compact and well separated. Du is defined as follows:

$$Du(K) = \min_{i=1,\ldots,K} \left(\min_{j=i+1,\ldots,K} \left(\frac{D(C_i, C_j)}{\max_{l=1,\ldots,K} d(C_l)} \right) \right) \tag{2.13}$$

where $d(C_i) = \max_{x,y \in C_i}(Dist(x,y))$ is the diameter of a cluster, $D(C_i, C_j) = \min_{x \in C_i, y \in C_j}(Dist(x,y))$ is the distance between clusters and K is the number of clusters. Large values of $Du(K)$ suggest the presence of compact and well-separated clusters.

2.5 Conclusion

In this chapter we presented basic concepts related to semantic layout generation. These concepts are fundamental for understanding the set of requirements introduced by our proposals, the operators and performance metrics used to assess state-of-the-art methods. The next chapter surveys a collection of related methods classified according to their purposes and functionalities.

CHAPTER 3

REVIEW OF THE LITERATURE

3.1 Introduction

In this chapter we describe a set of relevant methods to semantic layout generation. We organize the methods in six main categories according to their main characteristic, i.e. overlap removal, board-based, space-filling, alignment, hierarchy-based and dynamic update methods, which are detailed in the following and summarized in Table 1.

3.2 Overlap removal

Overlap removal techniques vary greatly as to mathematical formulation and they can be classified as physical-based, optimal-based and heuristic-based models.

Physical models such as force-based schemes (MISUE et al., 1995; HUANG et al., 2007; SPRITZER; FREITAS, 2012) and spring systems (CHUANG; LIN; YEN, 2004; HAREL; KOREN, 2002) are the methods used to position geometric entities representing graph nodes in the visual space. Overlap removal algorithms for layout arrangement and graph drawing differ as to the mathematical formulation as well as the nature of overlap removal problem. More precisely, graph drawing has an additional constraint related to the closeness (and distance) of nodes, that is, nodes have to be placed in the visual space such that the neighborhood of the nodes are preserved as much as possible. An issue with the physical-inspired overlap removal techniques is the lack of guarantees as to convergence and quality of the resulting layout. Such issue is addressed by optimal graph drawing techniques such as the ones proposed by Dwyer, Marriott e Stuckey (2006) and Marriott et al. (2003), both proposed energy functionals derived from intersection tests, which rely on quadratic optimization. In order to improve readability some techniques impose constraints to the energy functional towards generating grid-like arrangements (KIEFFER et al., 2013; RUEGG et al., 2014). The main problem with these optimal approaches is that the energy to be minimized depends on intersection tests, thus introducing several cases to be

Chart 1 – Classification of relevant to semantic layout generation methods by requirement fulfilled

Requirement	Class	Method
Overlap Removal	Physical Model	(MISUE et al., 1995)
		(HAREL; KOREN, 2002)
		(MARRIOTT et al., 2003)
		(CHUANG; LIN; YEN, 2004)
	Optimal base	(MARRIOTT et al., 2003)
		VPSC (DWYER; MARRIOTT; STUCKEY, 2006)
		PRISM (GANSNER; HU, 2009)
		(DWYER; MARRIOTT; STUCKEY, 2006)
		(KIEFFER et al., 2013)
		(RUEGG et al., 2014)
	Heuristic based	(DWYER; NACHMANSON, 2010)
		RWordle (STROBELT et al., 2012)
Board based	Small Multiples	(TUFTE, 1990)
		(BERTIN, 2007)
		(JAVED; MCDONNEL; ELMQVIST, 2010)
		(ELZEN; WIJK, 2013)
		(KEHRER et al., 2013)
		Correlated Multiples (LIU et al., 2013)
	Other approaches	IncBoard (PINHO; OLIVEIRA; LOPES, 2010)
		Self-Sorting Maps (STRONG; GONG, 2014)
		IsoMatch (FRIED et al., 2015)
Space Filling		Treemap (SHNEIDERMAN, 1992) and its variations
		(BUCHIN et al., 2011)
		NMap (DUARTE et al., 2014)
		Jasper (VALLET; MELANÇON; PINAUD, 2016)
Alignment		(XU et al., 2014), (XU et al., 2015)
		(JIANG et al., 2015)
Hierarchy based		Treemap (SHNEIDERMAN, 1992) and its variations
		(BEDERSON; SHNEIDERMAN; WATTENBERG, 2002)
		(BALZER; DEUSSEN; LEWERENTZ, 2005)
		(WOOD; DYKES, 2008)
		PedVis (TUTTLE; NONATO; SILVA, 2010)
		(NOCAJ; BRANDES, 2012)
Dynamic Update		ManiWordle (KOH et al., 2010)
		(REINERT; RITSCHEL; SEIDEL, 2013)

Source: Elaborated by the author.

handled by the algorithm. An interesting discussion about graph drawing literature can be found in (GANSNER; HU, 2009).

Heuristic methods also perform well in terms of overlap removal and similarity preservation. For instance, R-Wordle (STROBELT et al., 2012) can take into account information of similarity to place similar objects close to each other while avoiding overlaps. However, those methods do not make an efficient use of display area, mainly when dealing with geometric objects with different sizes. Scalability is another hurdle for those techniques, with a few exceptions (DWYER; NACHMANSON, 2010).

3.3 Board-based

These methods tiles the visual space with geometric primitives so as to produce a well structured layout. For instance, IncBoard (PINHO; OLIVEIRA; LOPES, 2010) arrange boxes into a regular grid. This arrangement reflects similarity between instances by proximity, i.e. neighbor cells in the grid represent similar instances in original space, allocated analogously to a chessboard where each square position is associated with an multidimensional instance. Self-Sorting Map (STRONG; GONG, 2011; STRONG; GONG, 2014) also relies on regular grids arrangement. Items are arranged based on a sorting mechanism using a hierarchical swapping process which aims to maximize normalized cross correlation between the input data and the target structured grid. Another method that is worth mentioning is IsoMatch (FRIED et al., 2015) that propose an optimization based on the error minimization of pairwise distances to allocate elements into a layout with a priori defined shape, such as, 2D grid and 3D sphere.

Small multiples (BERTIN, 2007; TUFTE, 1990) is another class of techniques that rely on structured arrangement of geometric primitives to build visualization layouts. Small multiples is typically used to create different views of a data set (ELZEN; WIJK, 2013) or to enable the visual analysis of multiple complex data such as time series (JAVED; MCDONNEL; ELMQVIST, 2010; KEHRER et al., 2013). Liu et al. (2013) proposed CorrelatedMultiples, a visualization method which makes use of Constrained Multidimensional Scaling (CMDS) to map into the 2D space a set of small multiples conveying similarity between them. A final process of alignment produce a structured and pleasant representation. However, it does not allow any interactive modification of the layout (as repositioning of instances), which is an important requirement on data exploration.

One of the main advantages of structured layouts is the ease interpretation of the resulting visualization. In fact, Elzen e Wijk (2013) demonstrated in their user study that visualizations made up of small multiples are more effective than other methods for exploring different views as well as orthogonal arrangement of the data. Similar conclusions have been reached by Javed, McDonnel e Elmqvist (2010) in the context of time series visual analytics. Despite the advantages, structured layouts not scale easily, impairing their use in applications involving large data sets.

3.4 Space-filling

These methods aim to maximize area usage of visual space. One of the best known methods in literature is Treemap. Treemaps were originally introduced by Shneiderman (1992) defined as spaced-constrained visualization of hierarchical structures. This approach is performed by recursively splitting the space in rectangular boxes whose size and orientation reflect the extent of nodes in the hierarchy. For a complete revision on treemap evolution along last decades see History of Treemap Research at the University of Maryland [1]. Most recent variations of Treemap incorporate remarkable features. For instance, Buchin et al. (2011) propose an optimization based on adjacencies among blocks – where each block represent a cluster of 2D rectangles – with purpose of transform an irregular layout into a spatial treemap, preserving cluster information as much as possible. Duarte et al. (2014) makes use of a slice-and-dice strategy for creating semantic preserved treemaps. Semantic information is defined by a multidimensional projection technique for embedding data into 2D space. Jasper (VALLET; MELANÇON; PINAUD, 2016) is a two-phase method for generating pixel-oriented layouts from large graphs. It summarizes into compact representations capable of conveying relevant information through a graph coarsening process. However, dynamic and incremental features are not present on its formulation, forcing to process the entire graph each time a node is modified.

3.5 Alignment

These methods are devoted to align horizontally/vertically geometric entities based on their spatial positions. For instance, Xu et al. (XU et al., 2014; XU et al., 2015) proposed important approaches for automatically inferring edge-alignment relationships. Such processes use an optimization mechanism for layout refinement in order to satisfy the inferred relationships while preserving as much as possible the original layout, in terms of position and size of geometric entities. Analogously, Jian et al. (JIANG et al., 2015) propose an algorithm for automatic generation of constraints using an energy function. The main difference against the method cited above, is a regularization step that resizes and rearranges elements.

It is worth to mentioning some important applications for these methods in the field of computer graphics to support the generation of urban procedural models (NISHIDA et al., 2016; MUSIALSKI et al., 2013) and engineering (MURPHY et al., 2012).

3.6 Hierarchy-based methods

These methods are designed to make an efficient use of display area while enabling dynamic navigation throughout different levels of a hierarchical representation (BLANCH; LECOLINET, 2007). Treemap (JOHNSON; SHNEIDERMAN, 1991) and its variants (BRULS;

[1] <http://www.cs.umd.edu/hcil/treemap-history/>

HUIZING; WIJK, 2000; FEKETE; PLAISANT, 2002; SHNEIDERMAN; WATTENBERG, 2001; SHNEIDERMAN, 1998-2013) are examples of hierarchy-based techniques tailored to visualize data organized as a tree structure. Another example is Pedvis (TUTTLE; NONATO; SILVA, 2010), a technique that builds upon H-tree layout and rectangular boxes to depict pedigree information structured as deep hierarchies. In contrast to the hierarchical methods described above, Voronoi Treemap (BALZER; DEUSSEN; LEWERENTZ, 2005) uses polygonal objects rather than rectangular boxes towards better representing the importance of nodes in the hierarchy and further differentiate sibling and non-sibling nodes. Despite the efficiency in space occupation, most tree-base layouts are not devised to place similar instances close to each other, making them unsuitable for applications involving similarity-based data exploration. Aiming at addressing this issue, some authors (BEDERSON; SHNEIDERMAN; WATTENBERG, 2002; WOOD; DYKES, 2008) have proposed ordering mechanisms that consistently arranges tree nodes according to some similarity measure. Nocaj e Brandes (2012) combined multidimensional scaling, Voronoi Treemap, and a set of visualization resources to highlight the similarity among data instances while enabling navigation throughout the hierarchy.

Hierarchy-based methods scale well and they make good use of display area. However, in contrast to structured arrangements, the layout resulting from those methods are not so easy to read as similar instances are not necessarily placed close to each other.

3.7 Dynamic update

A few methods tackle the problem of enabling interactive mechanism to enrich user experience during layout construction. ManiWordle (KOH et al., 2010) is one of the few examples in the context of textual data visualization, which allows users to interactively position words in the layout to dynamically update the overall arrangement. However, ManiWordle is not semantic aware, therefore, the task of building layouts where similar entities are placed next to each other falls entirely on the user's shoulder. Reinert, Ritschel e Seidel (2013) proposed a semantic aware fully interactive system that enables users to build customized layouts. Their approach is quite versatile and relies on a generalization of Centroidal Voronoi Tesselation to distribute primitives in the 2D space. The main drawback of Reinert et al.'s approach is the intricate energy functional that has to be evaluated and minimized, which demands tricky GPU implementations to ensure real-time layout handling.

3.8 Conclusion

In this chapter we summarize relevant methods to semantic layout generation. As can be noticed, most techniques are able to tackle just a few set of requirements simultaneously, impairing their use and flexibility. From chapter 4 to 7, we present in detail our proposed methods designed to account for multiple concurrent requirements such as object scale, neighborhood and aspect ratio preservation, optimal use of display area, and overlap removal.

CHAPTER

4

PROJSNIPPET

4.1 Introduction

Looking for useful information on Internet is daily task to millions of users. The traditional procedure consists in providing textual queries to a web search engine, which returns a ranked list of textual snippets each containing a content summary and a link to the referred web page. A ranked list of snippets is rather simple, straightforward to interpret, and it turns out to be effective in focused search tasks that require locating a particular document or web page (TEEVAN et al., 2009). Nonetheless, it also has limitations likely to hamper user experience when exploring and analyzing search results in other scenarios. Indeed, ranked lists fail to provide an overview of the collection retrieved, making it difficult and time consuming to figure out how documents relate contentwise. For instance, if a user queries Google's search engine on the keywords "jaguar features", the first page returned includes 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. Certainly users may refine the search, however, if s/he needs a global picture there is no other choice but navigating through the pages in the list and manually group the snippets according to their topic.

InfoVis offers users more flexible mechanisms to inspect and navigate 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, adding information on the contents of the returned documents. Although interesting and potentially useful, those visual resources reveal no information on document relations, that is, which documents share similar content and how many different subjects appear in the search results. Other types of methods replace the ranked list paradigm with alternatives such as thumbnails to favor better understanding of document content. Nonetheless, those methods tend to be visually more intricate and demand greater user effort to detect and inspect specific documents. Moreover, current visualization methods use the full content of each document, therefore relying in pre-processed data not readily made available by standard search engines, thus preventing their usage as independent plug-ins.

Multidimensional projection techniques are generally employed to generate visualizations that favor the perception of groups of similar documents. Such methods typically represent documents as points in a two-dimensional visual space, where neighboring points correspond to documents with similar content. However, points only convey information on neighborhood relations.

Superseding them with other geometric primitives introduces overlapping, which impairs readability. The problem of arranging geometric primitives in a two-dimensional visual space so as to ensure that similar objects stay neighbors while avoiding overlap is a recurrent problem in many different visualization contexts such as graph drawing, and word-cloud construction. Since finding an exact solution to the problem is computationally intractable, heuristics have been proposed, which though do not guarantee neighborhood preservation and visual space occupation. Therefore, techniques capable of arranging geometric entities in a visual space by taking into account their underlying object similarity while avoiding overlap are highly desirable, as they may benefit many distinct applications.

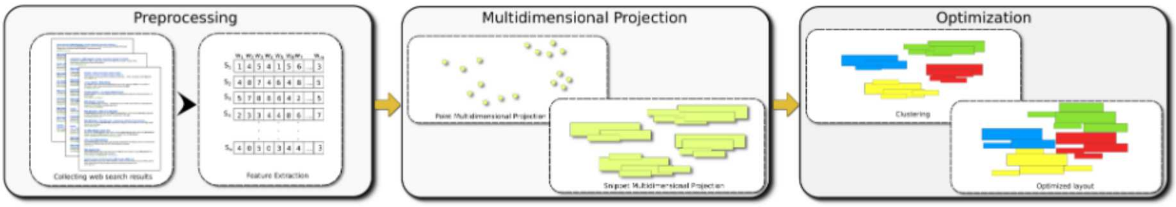
The technique described in this chapter provides an overlap removal approach that overcomes the drawbacks just discussed. More precisely, we propose an energy functional that considers both the overlapping between snippets and the neighborhood structure provided by a multidimensional projection. The minimum energy of such functional gives an arrangement of geometric entities in the visual space that preserves neighborhoods with minimum overlap.

We apply the proposed overlap removal approach in the context of snippet-based textual query web search result visualization, enabling two-dimensional layouts that preserve the simplicity and readability of textual snippets while emphasizing groups of content related 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 generated only from the information in the textual snippets, rendering the proposed method computationally efficient and easy to plug into conventional search engines.

4.2 The method

The proposed technique comprises three steps as shown in the pipeline in Figure 9: 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 (SALTON, 1991) for details on term frequency extraction). Stemming and stopword removal are applied and Luhn's lower and upper cuts (LUHN, 1958) established to compute the tf-idf vector representation of each snippet. Only the summary texts are processed, rather than the full content of the referred documents or web pages, which renders the visualization algorithm fast.

Figure 9 – 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.


Preprocessing
Collecting web search results
Feature Extraction
Multidimensional Projection
Point Multidimensional Projection
Snippet Multidimensional Projection
Optimization
Clustering
Optimized layout

Source: Gomez-Nieto et al. (2014).

Although considering the full document content might improve cluster quality, handling only the summary text favors interactivity and makes it easier to plug the proposed solution into existing standard browsers, which typically do not make available the full pre-processed content data.

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) (PAULOVICH et al., 2008) – 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. The k-means++ algorithm (ARTHUR; VASSILVITSKII, 2007) is applied to the projected layout to identify clusters of similar documents (using Euclidean distance as metric in the visual space), and rectangles in the same cluster may be assigned the same color to highlight groups. The benefits of highlighting clusters when visualizing and analyzing textual search results have been pointed out by several authors (CUTTING et al., 1992; PIROLLI et al., 1996), motivating our choice.

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 9) 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.

4.2.1 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:

$$E = (1-\alpha)E_O + \alpha E_N \tag{4.1}$$

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.

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 adjacent snippets. This is reached 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 enables resorting to simple and efficient optimization methods which are mandatory for quick generation of the final visualization.

Let $\mathbf{x},\mathbf{y} \in R^n$ be the coordinate vectors of the bottom left corner of each rectangle and $\mathbf{v},\mathbf{h} \in R^n$ 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

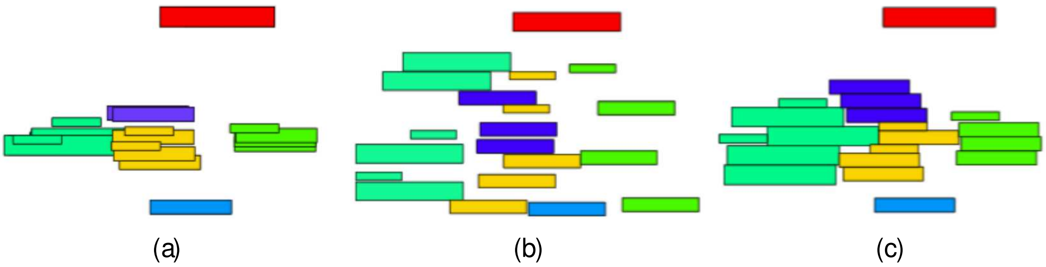
$$O_{i,j}(\mathbf{x},\mathbf{h}) = \begin{cases} \frac{1}{h_j^4}[h_j^2 - (x_i - x_j)^2]_+^2 & x_i \geq x_j \\ \frac{1}{h_i^4}[h_i^2 - (x_i - x_j)^2]_+^2 & x_i < x_j \end{cases}.$$

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 $O_{i,j}(\mathbf{x},\mathbf{h})$ 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., $O_{i,j}(\mathbf{y},\mathbf{v})$.

From definitions above we set E_O as:

$$E_O = \frac{2}{n(n+1)} \sum_{i=1}^{n} \sum_{j=i+1}^{n} \left[O_{i,j}(\mathbf{x},\mathbf{h}) O_{i,j}(\mathbf{y},\mathbf{v}) \right]. \tag{4.2}$$

Figure 10 – (a) Layout without optimization; (b) Overlapping energy E_O only; (c) both energies E_O and E_N combined ($\alpha = 0.5$).



Source: Gomez-Nieto et al. (2014).

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].

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:

$$l_{ij} = \begin{cases} -1/|i| & \text{if } j \neq i \text{ and } \overline{ij} \text{ is an edge of } G \\ 1 & \text{if } j = i \\ 0 & \text{otherwise} \end{cases} \tag{4.3}$$

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

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

$$\delta_x = L\mathbf{x}^0, \qquad \delta_y = L\mathbf{y}^0. \tag{4.4}$$

Notice that the components of δ_x and δ_y are given, respectively, by:

$$x_i^0 - \frac{1}{|N_i|}\sum_{j\in N_i} x_j^0, \qquad y_i^0 - \frac{1}{|N_i|}\sum_{j\in N_i} y_j^0 \tag{4.5}$$

In less mathematical terms, δ_x and δ_y measure how much each node deviates from the average of its neighbors. Therefore, we define the neighborhood energy as

$$E_N = \frac{n^2}{2\left(\|\delta_x\|_2^2 + \|\delta_y\|_2^2\right)}\left(\|L\tilde{x} - w\delta_x\|_2^2 + \|L\tilde{y} - w\delta_y\|_2^2\right). \tag{4.6}$$

It can be easily realized that E_N will be minimal when $\tilde{x}$ and $\tilde{y}$ are such that their differentials $L\tilde{x}$ and $L\tilde{y}$ are proportional to the initial differentials δ_x and δ_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 δ_x and δ_y).

The normalization factor $\frac{n^2}{2}\left(\|\delta_x\|_2^2 + \|\delta_y\|_2^2\right)^{-1}$ ensures 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.

Figure 10 illustrates the result of optimizing the layout shown in 10a. Figure 10b shows the layout produced by optimizing the overlapping energy only, whereas Figure 10c shows the outcome of the optimization procedure with both energy terms equally balanced.

4.2.2 Computational aspects and implementation

Horizontal and vertical bounds of the visualization windows are imposed as constraints for the minimization of the energy (4.1). This is necessary since for a sufficiently large positive number K, the coordinate vectors $\tilde{x} = K\tilde{x}^0$ and $\tilde{y} = K\tilde{y}^0$, ($w = K$) correspond to a global minimizer of E, as no overlap should appear 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:

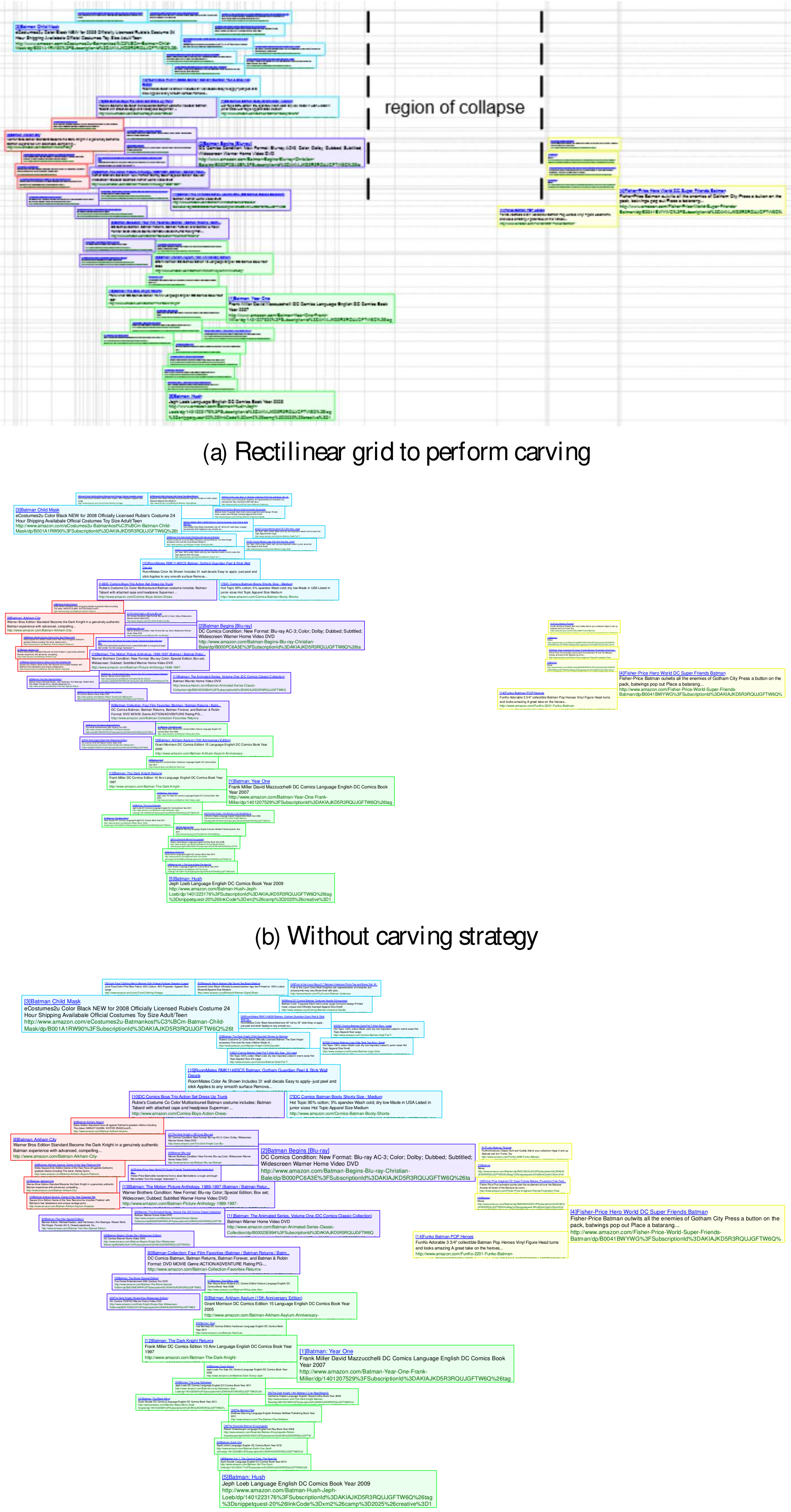
$$\begin{aligned} \min \quad & (1-\alpha)E_O + \alpha E_N \\ \text{such that:} \quad & x_{min} \leq x_i \leq x_{max} - h_i, \quad i = 1, \ldots, n \\ & y_{min} \leq y_i \leq y_{max} - v_i, \quad i = 1, \ldots, n. \end{aligned} \tag{4.7}$$

(recalling that variables x_i and y_i are encapsulated into E_O and E_N) which ensures that all rectangles lie within the visualization window, therefore preventing an exaggerated scaling effect.

The minimization is achieved by a globally convergent local optimization method, namely the Method of Moving Asymptotes (SVANBERG, 2002), available on the NLopt library at <http://ab-initio.mit.edu/wiki/index.php/NLopt>.

Reducing unused space For reducing unused space in the final layout we implemented a simplified version of the seam carving strategy (WU et al., 2011). The idea is to partition the

Figure 11 – Reducing unused space with a carving mechanism for the search term “batman”.



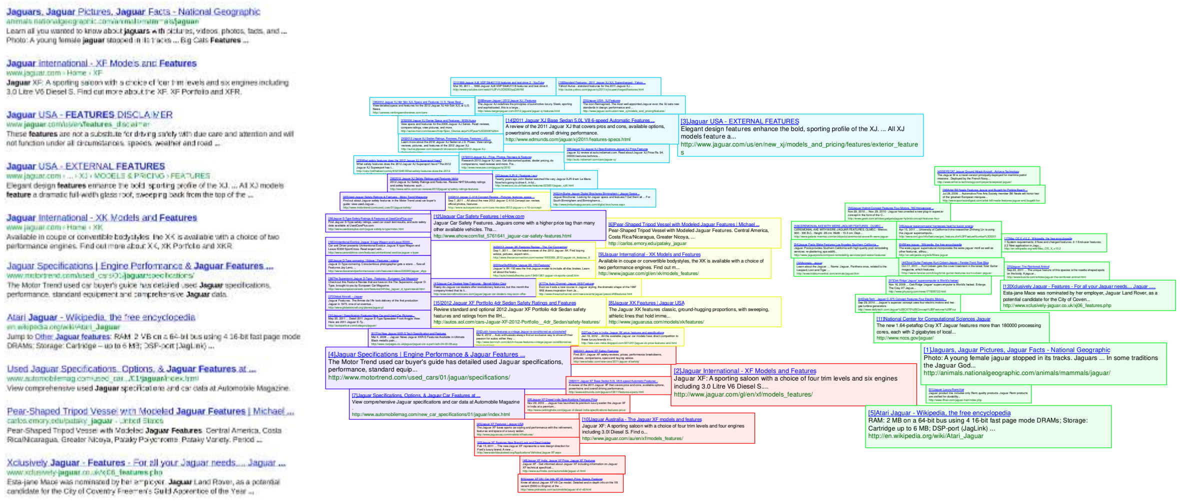
(a) Rectilinear grid to perform carving

(b) Without carving strategy

(c) With carving strategy

Source: Gomez-Nieto et al. (2014).

Figure 12 – Google and ProjSnippet views of the results of a query with terms “jaguar features”.



Jaguars, Jaguar Pictures, **Jaguar** Facts - National Geographic
animals.nationalgeographic.com/animals/mammals/**jaguar**
Learn all you wanted to know about **jaguars** with pictures, videos, photos, facts, and ...
Photo: A young female **jaguar** stopped in its tracks ... Big Cats **Features** ...

Jaguar International - XF Models and **Features**
www.jaguar.com › Home › XF
Jaguar XF: A sporting saloon with a choice of four trim levels and six engines including 3.0 Litre V6 Diesel S. Find out more about the XF, XF Portfolio and XFR.

Jaguar USA - **FEATURES** DISCLAIMER
www.**jaguar**.com/us/en/**features**_disclaimer
These **features** are not a substitute for driving safely with due care and attention and will not function under all circumstances, speeds, weather and road ...

Jaguar USA - EXTERNAL **FEATURES**
www.jaguar.com › ... › XJ › MODELS & PRICING › FEATURES
Elegant design **features** enhance the bold, sporting profile of the XJ. ... All XJ models **feature** a dramatic full-width glass roof, sweeping back from the top of the ...

Jaguar International - XK Models and **Features**
www.jaguar.com › Home › XK
Available in coupe or convertible bodystyles, the XK is available with a choice of two performance engines. Find out more about XK, XK Portfolio and XKR.

Jaguar Specifications | Engine Performance & **Jaguar Features** ...
www.motortrend.com/used_cars/01/**jaguar**/specifications/
The Motor Trend used car buyer's guide has detailed used **Jaguar** specifications, performance, standard equipment and comprehensive **Jaguar** data.

Atari **Jaguar** - Wikipedia, the free encyclopedia
en.wikipedia.org/wiki/Atari_**Jaguar**
Jump to Other **Jaguar features**: RAM: 2 MB on a 64-bit bus using 4 16-bit fast page mode DRAMs; Storage: Cartridge – up to 6 MB; DSP-port (JagLink) ...

Used Jaguar Specifications, Options, & **Jaguar Features** at ...
www.automobilemag.com/used_car.../01/**jaguar**/index.html
View comprehensive used **Jaguar** specifications and car data at Automobile Magazine.

Pear-Shaped Tripod Vessel with Modeled **Jaguar Features** | Michael ...
carlos.emory.edu/pataky_**jaguar** - United States
Pear-Shaped Tripod Vessel with Modeled **Jaguar Features**. Central America, Costa Rica/Nicaragua, Greater Nicoya, Pataky Polychrome, Pataky Variety. Period ...

Xclusively **Jaguar** - **Features** - For all your Jaguar needs.... Jaguar ...
www.xclusively-**jaguar**.co.uk/XC6_**features**.php
Esta-jane Mace was nominated by her employer, **Jaguar** Land Rover, as a potential candidate for the City of Coventry Freemen's Guild Apprentice of the Year ...

(a) First page of Google

(b) ProjSnippet displaying 64 snippets

Source: Gomez-Nieto et al. (2014).

“white regions” (snippet-free regions) of the visualization window into a rectilinear grid, as illustrated in Figure 11a. 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 fast and it obviously preserves the clusters. Although more sophisticated carving strategies exist capable of further removing unused space, they are computationally expensive and tend to spoil the neighborhood structures. The simple strategy described above is computationally efficient, produces pleasant layouts and preserves clusters altogether (see Figure 11).

4.3 Results, comparisons and evaluation

In the following we describe 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. All examples have been generated in a Intel Core^{TM}i7 CPU 920 2.66GHz with 8Gb of RAM. Stemming and stopword removal were applied in the pre-processing step, and Luhn’s lower and upper cuts were set to 12 and 33, respectively, to compute the tf-idf vector representation of the collection. A k-means++ clustering has been performed 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 applied with $\alpha = 0.3$ (the default value in our current implementation).

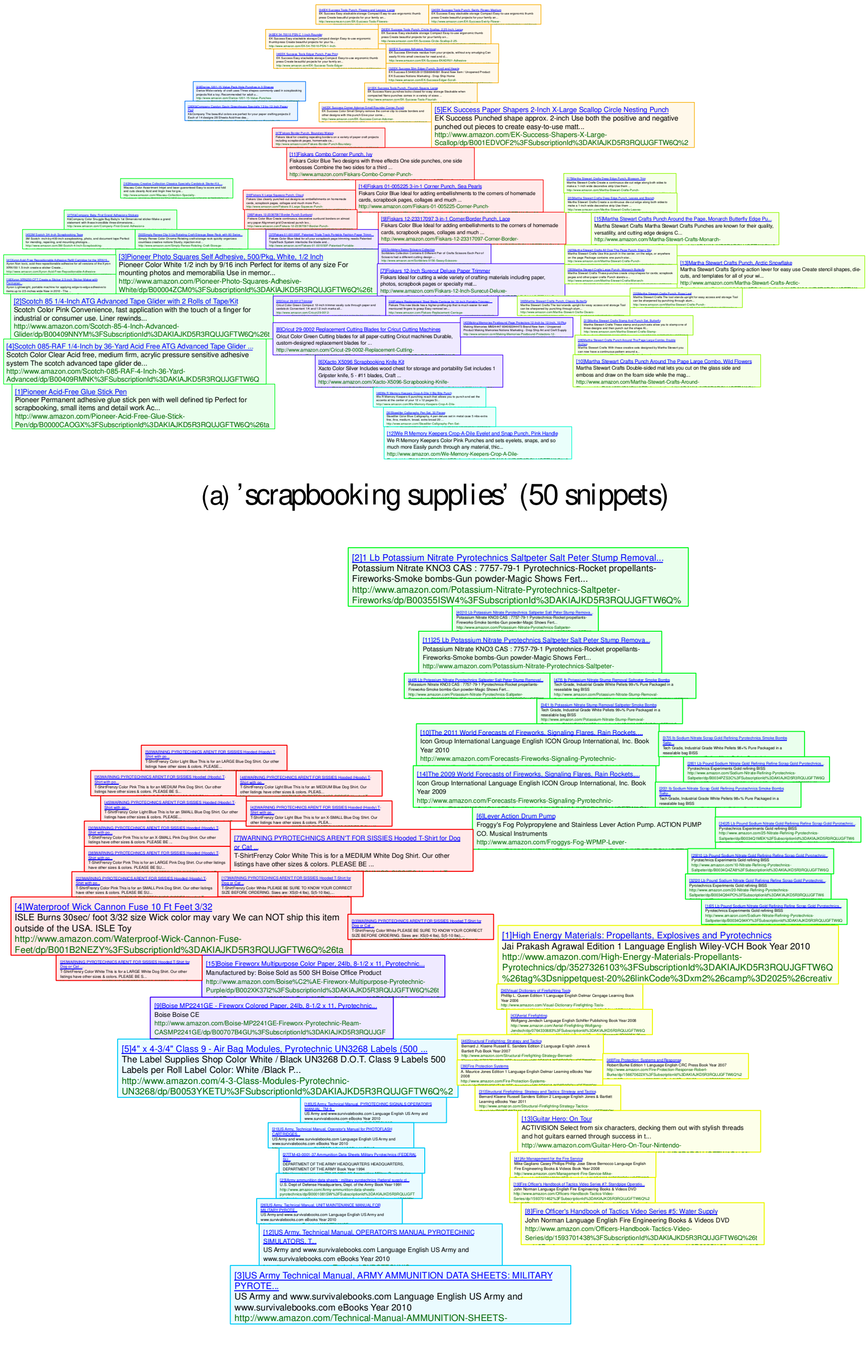
The first example illustrates a visualization displaying the results of a query on the terms “jaguar features” submitted to Google’s search engine. The view in Figure 12a shows the 10 best ranked snippets shown in the first page. Figure 12b displays a ProjSnippet view with the 64 best ranked snippets. 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.

Figure 13 shows the result of a search on Amazon's search engine, illustrating the potential of ProjSnippet in scenarios of searching for products at on-line stores. In these examples the fields Title, Author, Brand, Color, Edition, Feature, Published Language, Manufacturer, Product Group, Size, Warranty, Year of Publication we used to generate the vector space model.

In Figure 13a we issued a query with terms "scrapbooking supplies" in the category "Office Products", from which 50 products were returned and visualized. Overall, the layout organization reflects a global arrangement of the products by brand and functionality. Most of the snippets in the top orange group refer to punch models from the same brand, EK Success. There are also snippets that refer to an adhesive remover and a rounder, both products from the same brand as the punches. The red group includes only products from Fiskars, also comprising punches and corner and border punches. The light yellowish green group on the right displays products from a particular brand (Martha Stewart Crafts), again including mostly models of punches. The green group of snippets on the left is more varied in content, including different products from various brands. Still, the majority refers to various types of adhesives and related products: tape, tape gliders and tape refills; glue stick and varied occurrences of stickers, such as baby stickers and a sticker maker. The green group also includes a reference to cardstock and a reference to a craft storage rack. The central red group includes mostly references to utensils ranging from knife to cutting blades, from varied brands – including Fiskars, that has also utensils in the red group. The remaining two groups identified by the clustering are both very small. The two blue snippets on the top, to the left of the yellow group, refer to a hole puncher and a paper pad. The three ones in cyan at the bottom refer to punch models from a single brand and a pen set.

Figure 13b shows the visual result for a query on terms "pyrotechnic supplies" on

Figure 13 – Searching for “scrapbooking supplies” (a) and “pyrotechnics supplies” (b) on Amazon.


[5]EK Success Paper Shapers 2-Inch X-Large Scallop Circle Nesting Punch
EK Success Punched shape approx. 2-inch Use both the positive and negative punched out pieces to create easy-to-use matt...
[11]Fiskars Combo Corner Punch, Ivy
[3]Pioneer Photo Squares Self Adhesive, 500/Pkg, White, 1/2 Inch
Pioneer Color White 1/2 inch by 9/16 inch Perfect for items of any size For mounting photos and memorabilia Use in memor...
[2]Scotch 85 1/4-Inch ATG Advanced Tape Glider with 2 Rolls of Tape/Kit
Scotch Color Pink Convenience, fast application with the touch of a finger for industrial or consumer use. Liner rewinds...
[4]Scotch 085-RAF 1/4-Inch by 36-Yard Acid Free ATG Advanced Tape Glider ...
Scotch Color Clear Acid free, medium firm, acrylic pressure sensitive adhesive system The scotch advanced tape glider de...
[1]Pioneer Acid-Free Glue Stick Pen
Pioneer Permanent adhesive glue stick pen with well defined tip Perfect for scrapbooking, small items and detail work Ac...
[8]Cricut 29-0002 Replacement Cutting Blades for Cricut Cutting Machines
[6]Xacto X5096 Scrapbooking Knife Kit
[7]Fiskars 12-Inch Surecut Deluxe Paper Trimmer
[9]Fiskars 12-23117097 3-in-1 Corner/Border Punch, Lace
[14]Fiskars 01-005225 3-in-1 Corner Punch, Sea Pearls
[15]Martha Stewart Crafts Punch Around the Page, Monarch Butterfly Edge Pu...
[13]Martha Stewart Crafts Punch, Arctic Snowflake
[10]Martha Stewart Crafts Punch Around The Page Large Combo, Wild Flowers
[12]We R Memory Keepers Crop-A-Dile Eyelet and Snap Punch, Pink Handle

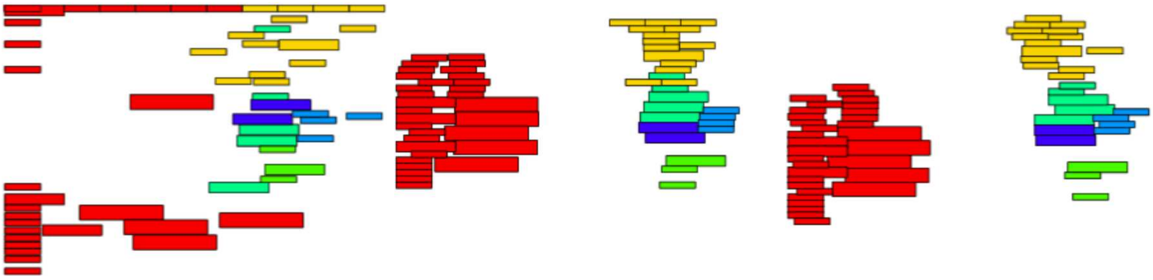
(a) ’scrapbooking supplies’ (50 snippets)

[2]1 Lb Potassium Nitrate Pyrotechnics Saltpeter Salt Peter Stump Removal...
Potassium Nitrate KNO3 CAS : 7757-79-1 Pyrotechnics-Rocket propellants-Fireworks-Smoke bombs-Gun powder-Magic Shows Fert...
http://www.amazon.com/Potassium-Nitrate-Pyrotechnics-Saltpeter-Fireworks/dp/B00355ISW4%3FSubscriptionId%3DAKIAJKD5R3RQUJGFTW6Q%
[11]25 Lb Potassium Nitrate Pyrotechnics Saltpeter Salt Peter Stump Remova...
[10]The 2011 World Forecasts of Fireworks, Signaling Flares, Rain Rockets...
[14]The 2009 World Forecasts of Fireworks, Signaling Flares, Rain Rockets...
[6]Lever Action Drum Pump
[7]WARNING PYROTECHNICS AREN'T FOR SISSIES Hooded T-Shirt for Dog or Cat ...
[4]Waterproof Wick Cannon Fuse 10 Ft Feet 3/32
ISLE Burns 30sec/ foot 3/32 size Wick color may vary We can NOT ship this item outside of the USA. ISLE Toy
[15]Boise Fireworx Multipurpose Color Paper, 24lb, 8-1/2 x 11, Pyrotechnic...
[9]Boise MP2241GE - Fireworx Colored Paper, 24lb, 8-1/2 x 11, Pyrotechnic...
[5]4" x 4-3/4" Class 9 - Air Bag Modules, Pyrotechnic UN3268 Labels (500 ...
[1]High Energy Materials: Propellants, Explosives and Pyrotechnics
[13]Guitar Hero: On Tour
[8]Fire Officer's Handbook of Tactics Video Series #5: Water Supply
[12]US Army, Technical Manual, OPERATOR'S MANUAL PYROTECHNIC SIGNALS, T...
[3]US Army Technical Manual, ARMY AMMUNITION DATA SHEETS: MILITARY PYROTE...

(b) ’pyrotechnics supplies’ (50 snippets)

Source: Gomez-Nieto et al. (2014).

Figure 14 – Effect of varying parameter α.



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

Source: Gomez-Nieto et al. (2014).

Amazon. The search, conducted in category All, returned 50 hits shown in the visualization. The red group on the left contains snippets referring to dog T-shirts designs with a printed phrase that includes the term "pyrotechnics". The exception is the bigger snippet that refers to a waterproof fuse. Just under is a blue cluster of just two snippets that refer to colored paper from a brand named Fireworx. Further down are the cyan snippets, most referring to US Army technical manuals on military pyrotechnics. Again, there is one exception, a reference to label supplies. The green cluster on the top right region refers mostly to chemical supplies, but it also includes two references to a book and a reference to a toy. Finally, the yellow group contains instruction material, mainly books but also video, on safety, addressing topics as protection, strategy, tactics and firefighting. Again, an exception is a reference to a video game. In this visualization the groupings are very uniform in content and clearly separated by topic, except for the few unusual items, such as the labels or the video game.

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. Moreover, the examples shown clearly illustrate that the ProjSnippet visualizations are capable of depicting many snippets simultaneously in a clear and organized manner.

The effect of varying the parameter α in equation (4.1) is illustrated in Figure 14, 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 Figure 14a 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 Figures 14b and 14c 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 Figure 14c) the snippets do not overlap unduly, showing the robustness of ProjSnippet as to the choice of α. No displacement of snippets occurs if $\alpha = 1$, since the result of the projection is clearly a minimizer of E_N.

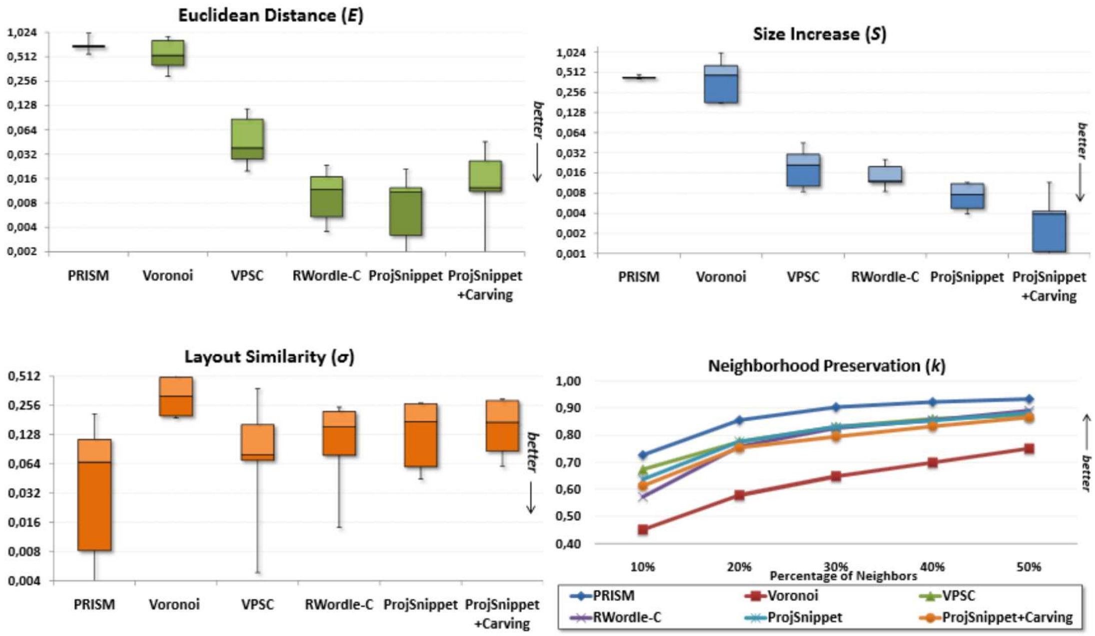
Table 1 shows the energy values after optimization, as well as computational times (in seconds) for the examples presented in the paper (the search "Batman" is depicted in Figure 11).

Table 1 – Optimization results.

Search	k	E_O	E_N	E	Time (s)
Pyrotechnics supplies	5	4.54^{-7}	1.97^{-4}	5.93^{-5}	0.24
Scrapbooking supplies	7	1.71^{-7}	5.09^{-4}	5.11^{-5}	0.23
Jaguar features	5	3.97^{-7}	1.91^{-4}	5.76^{-4}	0.27
Wave applications	6	5.38^{-6}	8.53^{-4}	3.44^{-4}	1.12
Batman	5	2.79^{-8}	4.86^{-5}	2.19^{-5}	0.26

Source: Gomez-Nieto et al. (2014).

Figure 15 – Comparing ProjSnippet, VPSC, PRISM, Voronoi-based, and RWordle-C considering metrics E (a), σ (b), S (c), and k-nearest neighborhoods (d). Better performance is indicated by lower values for E and S, values of σ closer to 1, and higher k-nearest neighborhood curves.



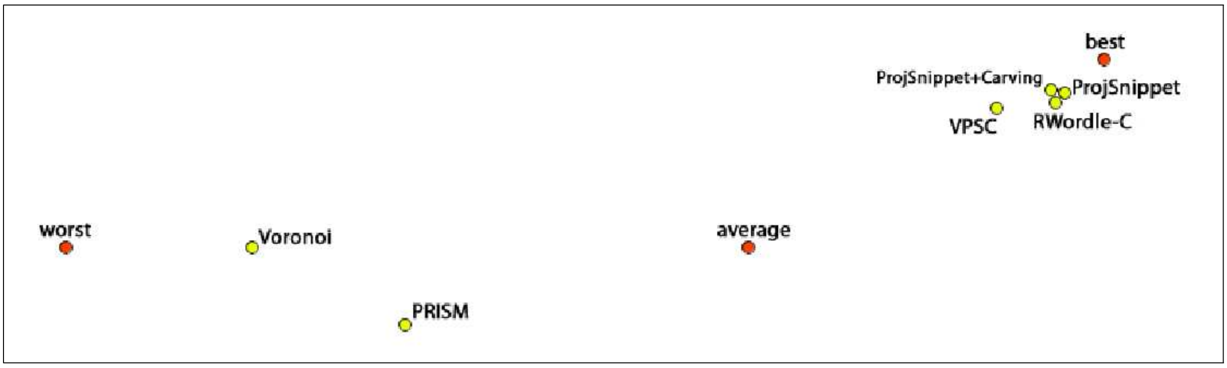
Source: Gomez-Nieto et al. (2014).

The minimization strategy does a pretty good job in quite acceptable times, supporting interactive visualization.

4.3.1 Comparing with overlap removal heuristics

Several heuristics have been proposed to arrange rectangular boxes in a viewport so as to avoid overlapping while still preserving the semantic relations among boxes as much as possible. In order to assess the effectiveness of the overlap removal mechanism built into ProjSnippet we have compared it with four well-known heuristics, namely, VPSC (DWYER; MARRIOTT; STUCKEY, 2006), PRISM (GANSNER; HU, 2009), Voronoi-based (DU; FABER; GUNZBURGER, 1999) and RWordle-C (STROBELT et al., 2012), regarding the following metrics: Euclidean Distance (Section 2.4.1), Layout Similarity (Section 2.4.2), Size Increase

Figure 16 – Global comparison of overlap removal methods regarding the four metrics simultaneously.



Source: Gomez-Nieto et al. (2014).

(Section 2.4.3) and Neighborhood Preservation (Section 2.4.5).

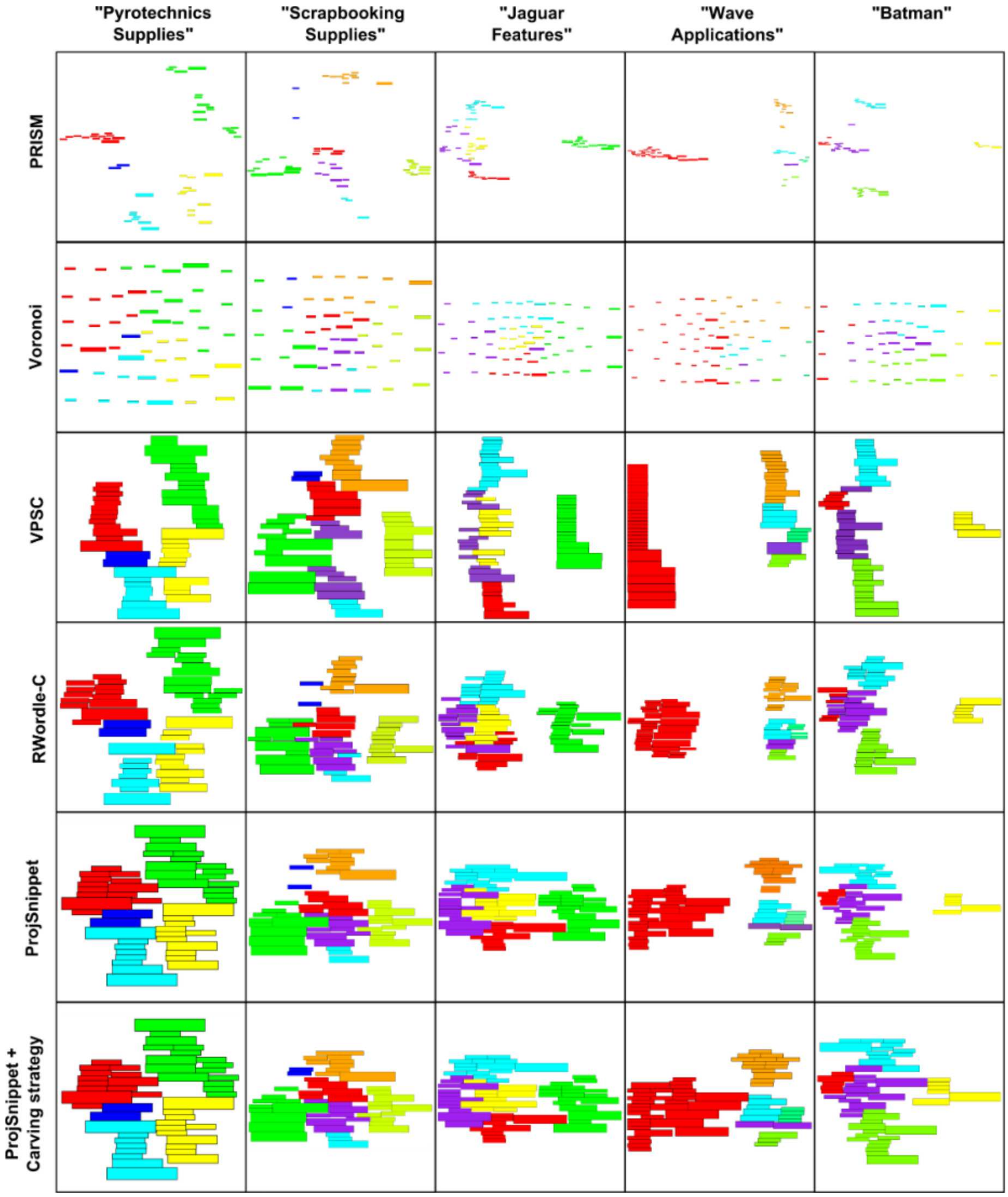
Figure 17 shows the layouts produced by the algorithms when applied to the data sets in the first column of Table 1. We run ProjSnippet with (ProjSnippet+Carving Strategy) and without (ProjSnippet) the seam-carving mechanism. One notices that ProjSnippet outputs a more organized layout, as compared with existing heuristics. On a first glance, its layouts resemble those obtained by RWordle-C, but ProjSnippet is more effective in preserving the grouping of similar elements (notice, for example, the red and the light blue groups on "Jaguar Features" and "Scrapbooking Supplies", respectively). Figure 15 summarizes the quantitative results by the above metrics: ProjSnippet performed quite well, resulting in better values than the other methods, in most cases.

Figure 16 shows a plot that considers all metrics simultaneously. Each overlap removal technique has been represented as a four-dimensional 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 labeled "best", "average", and "worst" in Figure 16 were created artificially as four dimensional vectors describing the best, average and worst results computed considering all methods over all data sets. More precisely, the coordinates of the point labeled "best" are given by the best value of each metric obtained in the experiments (over all data sets). The same for the "worst" point, now considering the worst values, whereas the "average" point is obtained by averaging the values of each metric computed from all methods over all data sets. The four dimensional vectors were projected with the LAMP multidimensional projection (JOIA et al., 2011). The techniques closer to "best", namely ProjSnippet and RWordle-C, are the ones with the best global performance, relative to all metrics.

4.3.2 User evaluation

We conducted two controlled user evaluations: one comparing ProjSnippet with a standard list-based interface and another comparing it against other layout techniques, namely PRISM, VPSC and RWordle-C. The first study was aimed at assessing whether the ProjSnippet

Figure 17 – Layouts produced by ProjSnippet, VPSC, PRISM, Voronoi-based, and RWordle-C for five distinct data sets.



Source: Gomez-Nieto et al. (2014).

layout allows users to find information faster than a list-based interface in tasks that require identifying groups of related sites, without significantly affecting precision. The second study was aimed at comparing ProjSnippet with other layout techniques, regarding the correctness of such tasks.

We formulated specific questions, detailed in Table 2, relative to the two queries already introduced, on "pyrotechnics supplies" (DT1) and on "jaguar features" (DT2). Each snippet in the interfaces shows its rank as returned by the search engine, so that the rank could be taken as a site identifier by subjects answering the questions, when required.

Both evaluations followed the same overall procedure comprised of four steps:

1. Introduction: participants were given a brief explanation on the purposes of the study.

2. Tool exposure: participants were shown basic functionalities and interaction functions of the prototypes interfaces (ProjSnippet and list-based).

3. User familiarization: participants interacted with their relevant interfaces for around 10 minutes, exploring a collection other than DT1/DT2.

4. Evaluation: participants were invited to answer the questions in Table 2 on their assigned interface/collection.

For the first study we invited 14 participants, all undergraduate or graduate students to execute the tasks using ProjSnippet and the standard list-based interface. Subjects were split into two groups of seven, so that a group used the list-based interface to answer questions on the "pyrotechnics supplies" hits and the ProjSnippet interface to answer questions on the "jaguar features" hits, whereas the other group used the interfaces in the reverse order. This study verified the following hypothesis:

- H: Users of ProjSnippet will spend less time to answer questions that require a global analysis of the query results (T1, T2, T3 and T4), with no significant loss in correctness.

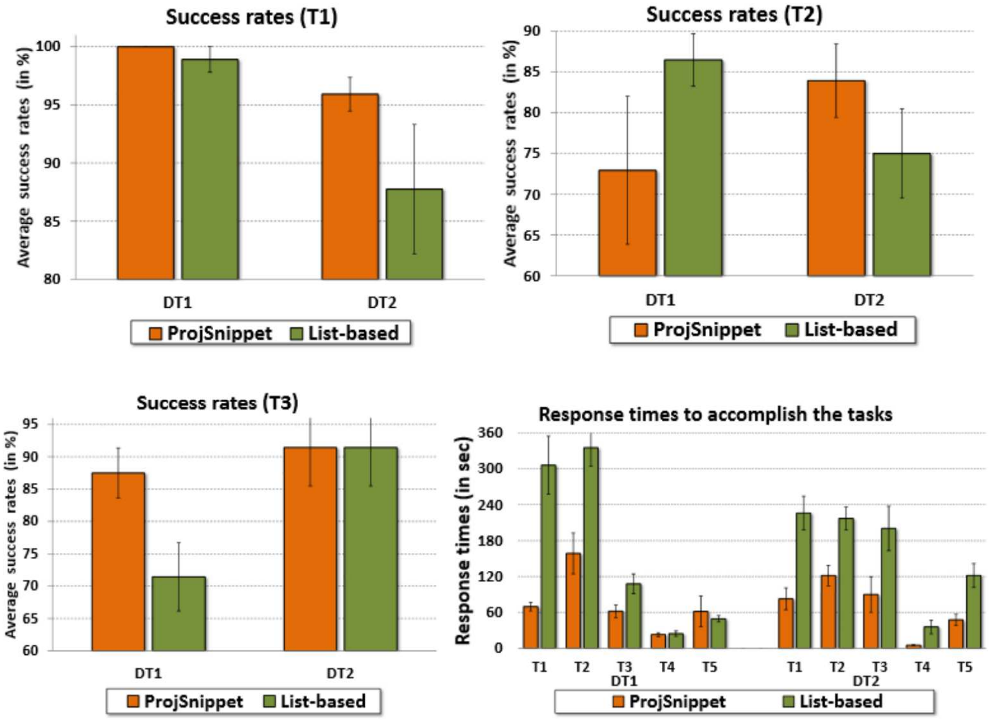
We measured correctness of the answers (success rates) and the elapsed times taken to answer questions T1 to T5. In order to assess only the effectiveness of the proposed two-dimensional layout we turned off the clustering mechanism, that is, clusters of similar snippets have not been colored or highlighted. For the sake of fairness we also disabled the carving mechanism available in ProjSnippet. Results are shown in Figure 18, for each task and each dataset. 100% success rates were achieved on both collections on tasks T4 and T5. We applied a T-test with a 5% level ($\alpha = 0,05$) to check for statistical significance of the differences found.

One observes in Figure 18(a) that subjects answering task T1 ("how many websites...") achieved better correctness on the ProjSnippet interface, on both collections. The difference, Chart 2 – Task questions in user tests.

	Task Target	Question
T1	Identify groups of related sites	DT1: How many websites report on chemical supplies for pyrotechnics?
		DT2: How many sites depict content on the “Jaguar XJ” car model?
T2	Identify groups of related sites	DT1: Which sites present books, guides or papers about pyrotechnics?
		DT2: Which sites depict content on the “Jaguar XK” car model?
T3	Identify groups of related sites	DT1|DT2: How many different topics you can identify in the returned results?
T4	Find different sites addressing similar content	DT1: Find three websites announcing “T-Shirt for Dogs”.
		DT2: Find two websites that refer to the animal “Jaguar”.
T5	Find a particular site	DT1: Find a website that addresses the topic “wick cannon fuse”.
		DT2: Find a website that includes the expression “Jaguar Features”.

Source: Gomez-Nieto et al. (2014).

Figure 18 – Correctness (average success rate) of subject answers in Tasks T1, T2 and T3 when using the ProjSnippet and the List-based interfaces, and the response times, in seconds, to all tasks.



Source: Gomez-Nieto et al. (2014).

Table 2 – ProjSnippet x List-based T-test p-values

Task	"pyrotechnics supplies" (DT1)	"jaguar features" (DT2)
T1	0.002629	0.001074
T2	0.002387	0.002976
T3	0.036768	0.039171
T4	0.652960	0.005878
T5	0.780640	0.036399

Source: Gomez-Nieto et al. (2014).

Table 3 – ANOVA p-values relative to comparison of the four layout techniques.

Task	"pyrotechnics supplies" (DT1)	"jaguar features" (DT2)
T1	0.2427	0.5827842
T2	0.426798	0.020015
T3	0.012762	0.908402

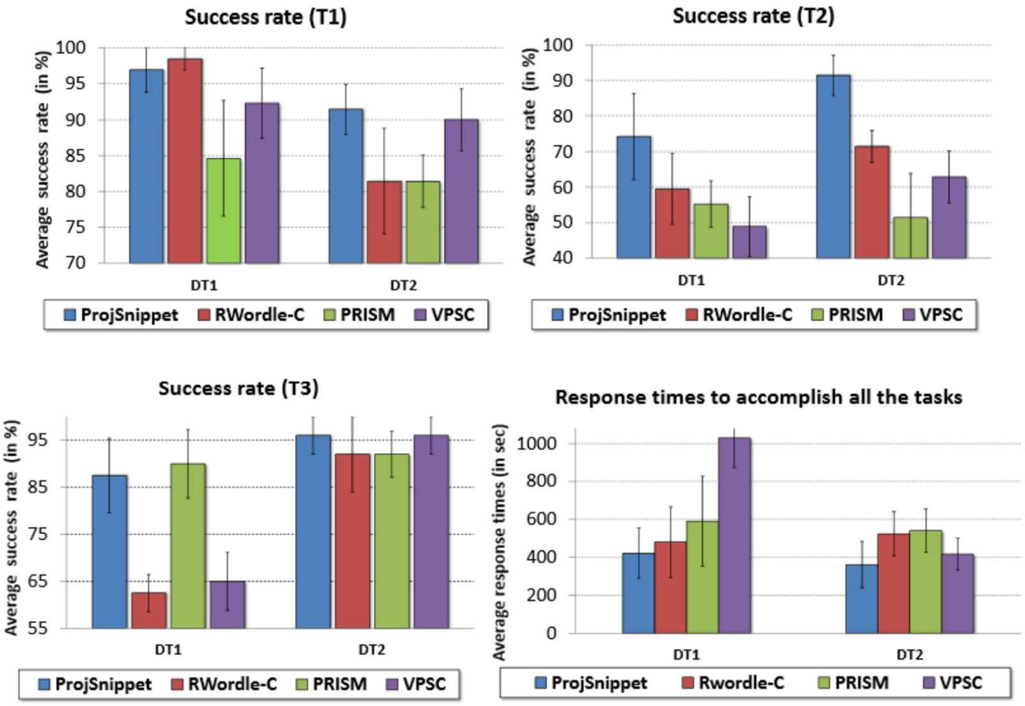
Source: Gomez-Nieto et al. (2014).

however, is not statistically significant. Participants answering task T2 ("which websites...") performed better on the list-based interface on DT1 ("pyrotechnics supplies"), and better with ProjSnippet on DT2 ("jaguar features"). Again, differences have not been found to be statistically significant. Finally, on task T3, which required identifying the multiple topics addressed, performance of ProjSnippet users was equivalent to those of the list-based on DT2, and better on DT1 – the only difference found to be of statistical significance. Therefore, we conclude that in general users could identify the relevant sites with both interfaces. In fact, in most cases users of ProjSnippet performed better, albeit it is not possible to conclude that it favors an improvement in the success rates.

Figure 18(d) confirms that ProjSnippet users took less time to answer all questions on both collections, with one single exception (task T5 on DT1). Differences have been found to be statistically significant for tasks T1, T2 and T3 on DT1 ("pyrotechnics supplies") and for all five tasks on DT2 ("jaguar features"). Table 2 shows the p-values computed for the time differences in tasks, on both collections. Even for task T5 (identifying a particular website), ProjSnippet users performed better on DT2, whereas we expected scanning through the list view would be faster. These results confirm our initial hypothesis.

For the second user study we invited 24 participants, again undergraduate or graduate students in Computer Science and none of them involved in the previous study. They were asked to answer the same questions, with evaluation taking place in two stages: first, each subject worked on DT1 displayed by a particular layout technique, and then on DT2 and a different layout technique. Subjects were initially randomly assigned to 4 groups, and each group of six assessed one layout technique. In the second stage subjects were reassembled into 4 groups ensuring they would work on a layout technique different from the previous one. The working hypothesis can be stated as:

Figure 19 – Correctness (average success rate) of subject answers in Tasks T1, T2 and T3 for the different layout techniques, and overall response times (in seconds).



Source: Gomez-Nieto et al. (2014).

- H: Users of ProjSnippet will achieve better success rates than users of other layouts when answering questions that require a global analysis of the query results (T1, T2 and T3).

Subjects spent roughly 20 to 30 min to complete each stage. We measured correctness of the answers to questions T1 to T5. Again, 100% success rates were achieved on tasks T4 and T5 on both collections, and results for tasks T1, T2 and T3 are shown in Figure 19. We applied one-way ANOVA at 5% level to check for statistical significance of the performance differences, corresponding values are shown in Table 3.

Analysis of Figure 19 reveals that ProjSnippet users did better than the others on task T2, on both datasets. However, only in DT2 the performance difference was found to be significant. They also did better on Task T1 with DT2, whereas with DT1 the ProjSnippet layout came as second best. Again, differences have not been found to be statistically significant. On task T3 ProjSnippet came second to PRISM on DT1, and first (but equivalent to VPSC) on DT2, with statistical significance found on DT1 only.

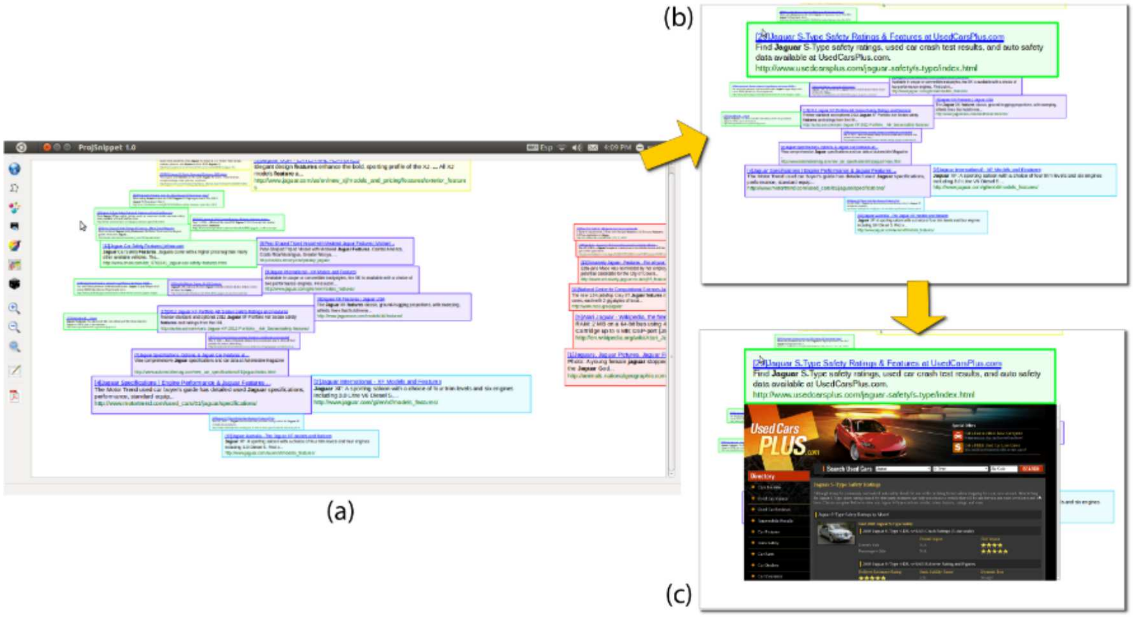
Although these results do not allow us to confirm the original hypothesis on the superiority of ProjSnippet, we observe it displayed good performance and a more stable behavior across tasks than the other techniques considered. Its users achieved higher or equivalent average success rates, as compared to users of other layouts, in four out of the six scenarios, and they also did well in the remaining two. They also spent less time executing their tasks, as observed in Figure 19(d).

Notice that Figure 16 indicates the inferior quality of PRISM compared to ProjSnippet, RWordle-C, and VPSC regarding the quantitative measures, which is consistent with the observations from our user evaluation, where PRISM users did worse than users of the other three techniques in all except one task (T3/DT1). Moreover, RWordle-C users performed better in one evaluation (T1/DT1), being second in other two (T2/DT1 and T2/DT2), while VPSC tied with ProjSnippet in the first position once (T3/DT2), being second in the T1/DT2 test. Based on these evidences one could claim that RWordle-C has a better performance than VPSC, again in agreement with Figure 16. We point out, though, that additional studies should be conducted to further investigate the relationship between quantitative measures provided by the metrics and the qualitative results resulting from our user evaluation.

4.4 Discussion and limitations

The ProjSnippet views of a collection of returned hits highlight their global relationships, as opposed to organizing them by their inferred relevance to the query. Still, the visualizations retain the simplicity of the snippet-based interaction, which from our perspective is a significant advantage. The underlying visualization paradigm is modified gently, requiring no substantial additional effort from users familiar with the standard list-based views. Even the aspect ratio of the rectangles reflects the nature of textual snippets, which are wider than higher. Users can still navigate the snippets and click to see a web-page preview (see Figure 20) and to inspect the

Figure 20 – Interactive exploration with ProjSnippet: (a) Main window, (b) on mouse over a snippet is highlighted and enlarged, (c) after a few seconds a preview of the page content is displayed.



[26]Jaguar S-Type Safety Ratings & Features at UsedCarsPlus.com
Find **Jaguar** S-Type safety ratings, used car crash test results, and auto safety data available at UsedCarsPlus.com.
http://www.usedcarsplus.com/jaguar-safety/s-type/index.html

Source: Gomez-Nieto et al. (2014).

contents of particular documents.

Both the display size and the overall number of snippets exhibited affect visualization readability. The illustrative examples shown were handled on medium to large-sized monitors and were readily interpreted and easily read. Obviously, readability will be hampered on small monitors, in which case it is better to display less snippets. Finding an optimal number of snippets to display is not straightforward, since a decision involves many variables, such as the screen resolution and the nature of the search. Moreover, if the user-defined number of clusters is not set properly, non-similar snippets may end up in the same cluster and mislead user interpretation.

Our examples also indicate that creating the visualizations only from the summarized snippet texts is quite satisfactory. Similar entities are nicely clustered, although some apparent "outliers" may occur. Cluster quality might be further improved by inputting additional text from the documents into the clustering algorithm. Nonetheless, this would incur in higher computational cost and not necessarily produce better results, as text clustering is intrinsically fuzzy: in many situations one could easily justify assigning a document to multiple clusters.

ProjSnippet requires a very simple pre-processing step, but some tricky issues remain. For example, setting appropriate values for Luhn's lower and upper cuts in scenarios where little information is available, as it is the case here, is not straightforward and deserves further investigation. In our examples we typically employed a lower cut of three and no upper cut and removed the query terms from the vector representation.

Finally, clustering in visual space will produce good results as long as the projection technique does a good job of preserving the relevant neighborhoods. Our choice of the LSP method is justified by its being known to perform quite well in terms of neighborhood preservation, which is confirmed by the results shown here. Nonetheless, if this is not the case, the visualization of groups may be misleading. Users may investigate alternatives playing with the number of clusters while observing the visualization.

4.5 Conclusion

In this chapter we described ProjSnippet, a technique to visualize the collection of textual snippets returned from a web query. This approach builds intuitive and meaningful layouts that optimize the placement of snippets by employing an 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.

CHAPTER

5

MIOLA

5.1 Introduction

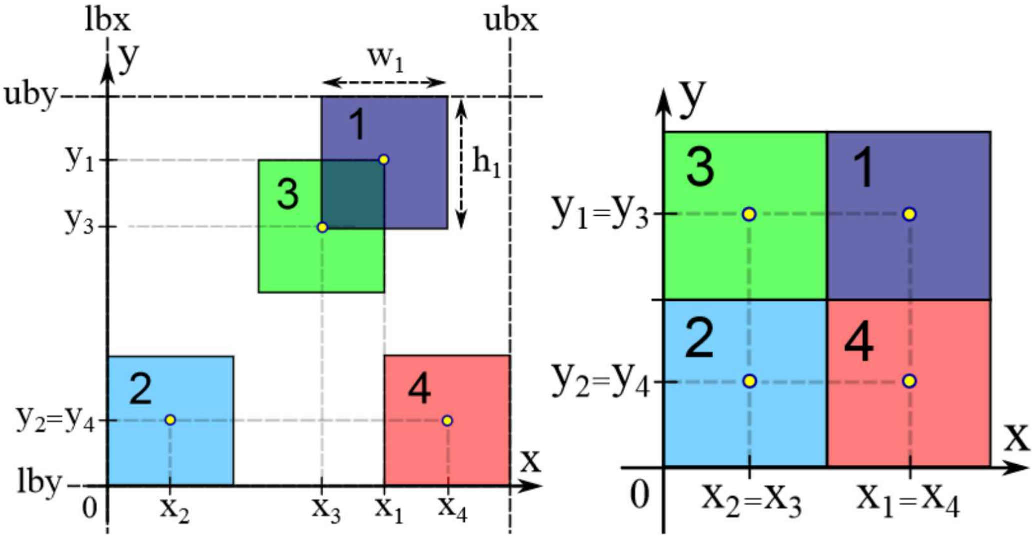
In this chapter we describe Mixed Integer Optimization for Layout Arrangement (MIOLA), a technique to tackle the problem of arranging rectangular boxes in the visual space so as to place objects representing similar content close to each other while avoiding overlaps. We formulate the problem as a Mixed 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.

The practical usefulness of our method is demonstrated in a video data set visualization application. Textual information associated to each video is used to measure the similarity between them, that is, the similarity between videos is calculated using bag-of-words derived from textural information and the cosine distance measure. The similarity measure is, then, used by a multidimensional projection technique that maps the data to the visual space, where the boxes representing the videos are arranged by the proposed optimization mechanism, generating the final layout.

5.2 The method

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 point, 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

Figure 21 – Inequalities (5.2) ensure that no boxes will not displayed offscreen while inequalities (5.3) preserve the order of the coordinates w.r.t. initial layout. (a) Input layout and (b) result after layout arrangement (orthogonal ordering: $x_{p_i} \le x_{p_{i+1}}$, $p_1 = 2, p_2 = 3, p_3 = 1, p_4 = 4$; similarity for y case).



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

Source: Gomez-Nieto et al. (2013).

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.

5.2.1 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 $B_i = (x_i, y_i, w_i, h_i) \in R^4$, where (x_i, y_i), $w_i > 0$, $h_i > 0$ are the centroid, width and height of B_i, respectively (see Fig. 21(a)). Two boxes B_i and B_j do not overlap if and only if one of the following inequalities holds:

$$|x_j - x_i| \ge \frac{w_i + w_j}{2} \quad \text{or} \quad |y_j - y_i| \ge \frac{h_i + h_j}{2} \tag{5.1}$$

We refer to the inequalities in (5.1) as non-overlap constraints.

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

$$\begin{aligned} \frac{w_i}{2} + lbx \le x_i \le ubx - \frac{w_i}{2} \quad \text{(x lower/upper bounds)} \\ \frac{h_i}{2} + lby \le y_i \le uby - \frac{h_i}{2} \quad \text{(y lower/upper bounds)} \end{aligned} \tag{5.2}$$

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 (5.1) and (5.2) 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,

$$\begin{aligned} x_{p_1} \leq x_{p_2} \leq \ldots \leq x_{p_n} \ (x \text{ orthogonal order}) \\ y_{q_1} \leq y_{q_2} \leq \ldots \leq y_{q_n} \ (y \text{ orthogonal order}) \end{aligned} \tag{5.3}$$

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. 21).

Therefore, by moving the centroid of the boxes while ensuring Equations (5.1), (5.2), and (5.3) can generate an overlap free layout that preserves the initial neighborhood structures.

5.2.2 The MIQP formulation

The problem of positioning the boxes B_i in the visual space so as to ensure that Equations (5.1), (5.2), and (5.3) hold can be formulated as a Mixed Integer Quadratic Programming Problem (LAZIMY, 1982) as follows:

$$\begin{aligned} &\min_z \quad f(z) = \frac{1}{2} z^T Q z = \sum_{i=1}^{n} \sum_{j=i+1}^{n} \mathrm{Dist}^2(B_i, B_j) \\ &\text{subject to} \quad \begin{cases} Az \leq b \\ lb \leq z \leq ub \end{cases} \\ &z = \begin{bmatrix} x \\ y \\ r \end{bmatrix}, \quad \begin{aligned} x &= (x_1, x_2, ..., x_n)^{>} \in R^n \\ y &= (y_1, y_2, ..., y_n)^{>} \in R^n \end{aligned} \\ &r = (r_{12}, \ldots, r_{1n}, r_{23}, \ldots, r_{2n}, \ldots, r_{n-1n})^{>}, \ r_{ij} \in \{0, 1\} \end{aligned} \tag{5.4}$$

where z is the sought solution, $\mathrm{Dist}(B_i, B_j)$ denotes the euclidian distance between (x_i, y_i) and (x_j, y_j), and vectors lb (lower bounds) and ub (upper bounds) are the visualization window bounds as defined in the inequalities (5.2). Q is the positive semi-definite matrix composed by blocks L given by:

$$Q = \left[\begin{array}{c:c} \begin{array}{c:c} L & 0 \\ \hdashline 0 & L \end{array} & 0 \\ \hdashline 0 & 0 \end{array}\right], \quad L = nI_d - \mathrm{ones}(n, n),$$

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 b are defined so as to incorporate the constraints (5.1) and (5.3). Precisely, the ordering given by (5.3) allows us to write:

$$x_{p_1} \leq x_{p_2} \leq \ldots \leq x_{p_n} \Rightarrow x_{p_i} - x_{p_{i+1}} \leq 0, \ i = 1 \ldots n-1, \tag{5.5}$$

with a similar expression holding for y. Additionally, the sorting also allows us to get rid of the absolute value function in (5.1), that is,

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

$$x_{p_i} - x_{p_j} \leq -\frac{(w_{p_i} + w_{p_j})}{2}, \ i < j, \tag{5.6}$$

with a similar expression holding for y. The variables r_{ij} allows for incorporating the OR condition defined in (5.1) into the optimization problem as follows:

$$\begin{gathered} x_{p_i} - x_{p_j} \leq w_{ij} + Mr_{ij} \Leftrightarrow y_{q_i} - y_{q_j} \leq h_{ij} + M(1 - r_{ij}), \\ i < j, \ r_{ij} \in \{0, 1\}, \end{gathered} \tag{5.7}$$

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 (5.7) is that if $r_{ij} = 0$, the constraint $x_{p_i} - x_{p_j} \leq w_{ij}$ becomes mandatory while $y_{q_i} - y_{q_j} \leq h_{ij} + M$ 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 (5.1), (5.2), and (5.3).

Equations (5.5) and (5.7) are incorporated into matrix A using the auxiliar matrices C_x, C_y, D_x, D_y as follows:

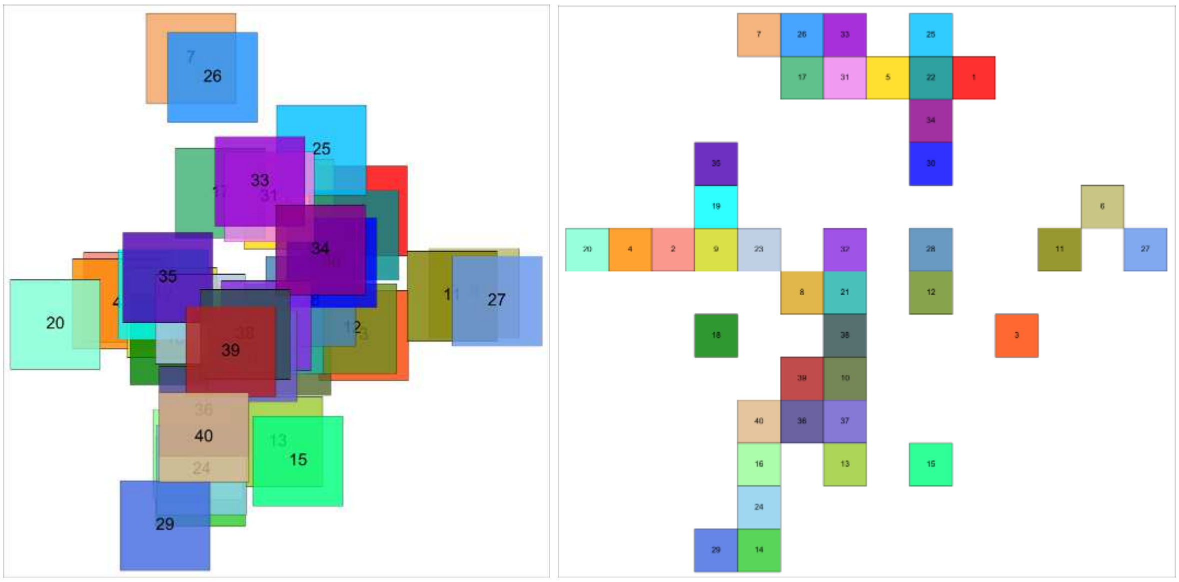
$$A = \begin{bmatrix} \begin{bmatrix} C_x & 0 & 0 \\ 0 & C_y & 0 \end{bmatrix} \\ \hline \begin{bmatrix} D_x & 0 & -MI_d \\ 0 & D_y & MI_d \end{bmatrix} \end{bmatrix} \quad b = \begin{pmatrix} c \\ \hline d \end{pmatrix} \tag{5.8}$$

where $[C_x|C_y|c]$ and $[D_x|D_y|d]$ are built from constraints (5.5) and (5.7), respectively. Notice that vector c is null as stated in (5.5). However, this constraint can be relaxed so as to introduce more flexibility into the layout, as explained in subsection 5.2.3. Figure 22 illustrates the resulting layout from solving the standard MIQP (5.4). Notice that the orthogonal order is accurately preserved while the boxes are thoroughly spread in the visualization window.

5.2.3 Relaxation of constraints

Besides providing accuracy and robustness when dealing with complex layouts, the formulation (5.4) enables great flexibility in generating a variety of non-overlap arrangements.

Figure 22 – Result of our method to rearrange boxes with overlaps.



(a) Input layout.

(b) Optimized layout.

Source: Gomez-Nieto et al. (2013).

Different layouts can be easily obtained by introducing relaxation constants within lower lb and upper ub bounds in Equations (5.4) and/or vector $b = [c|d]^T$ in (5.8) as follows:

$$lb_{relax} = lb + \overline{lb}, \quad ub_{relax} = ub + \overline{ub} \tag{5.9}$$

$$c_{relax} = c + \overline{c}, \quad d_{relax} = d + \overline{d}, \tag{5.10}$$

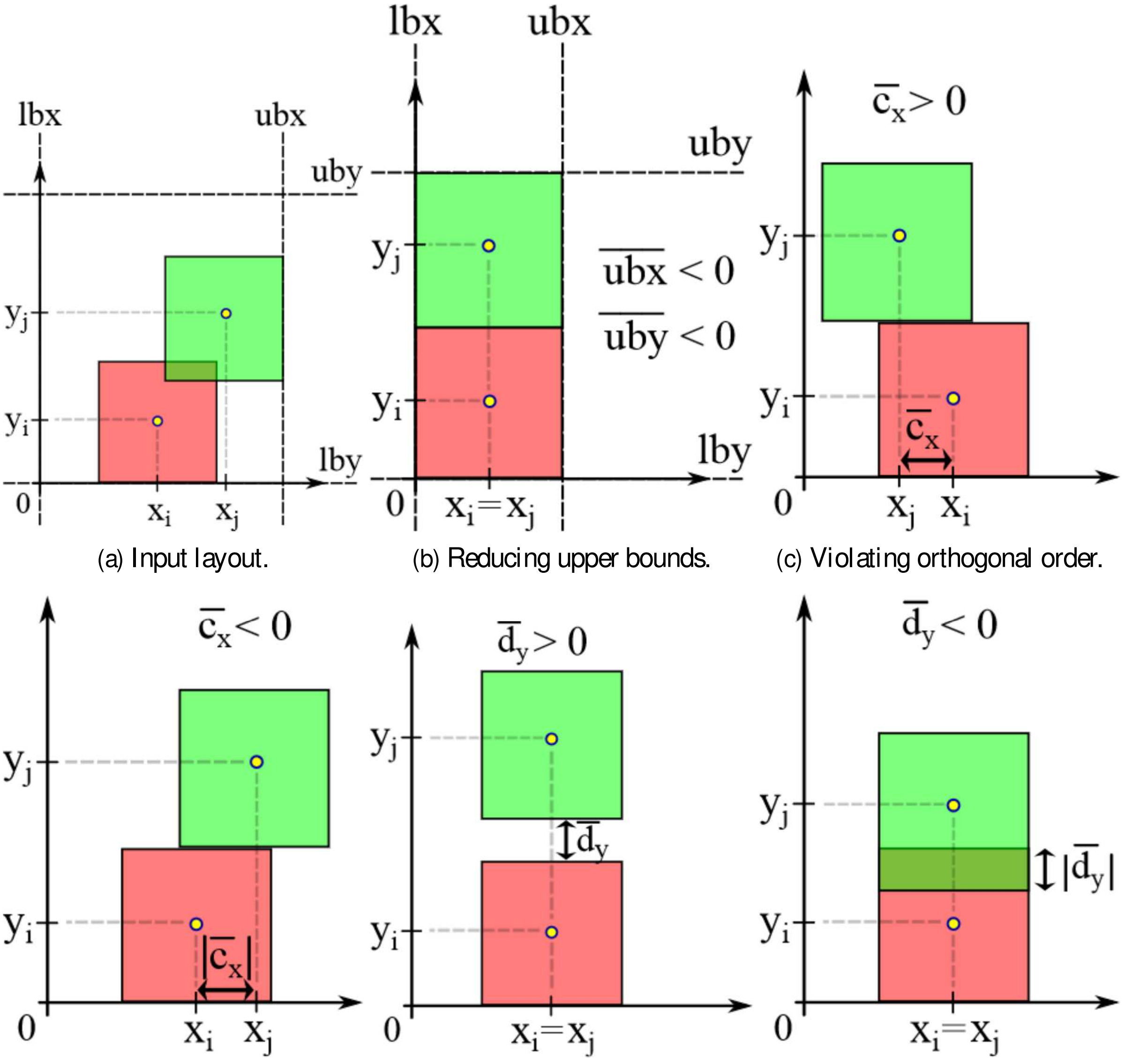
where $\overline{lb} = (\overline{lb_i})$, $\overline{ub} = (\overline{ub_i})$, $\overline{c} = (\overline{c_k})$ and $\overline{d} = (\overline{d_l})$ are the relaxation constants. Notice that relaxation constants can be individually chosen for each box in (5.9) or for each pair of boxes defined in (5.10). Moreover, each coordinate of $\overline{lb}$, $\overline{ub}$, $\overline{c}$ and $\overline{d}$ can assume both positive and negative values.

While $\overline{lb}$ and $\overline{ub}$ regulate the effective visual space of the boxes, $\overline{b}$ and $\overline{c}$ control the relaxation distances of orthogonal ordering and non-overlap constraints. Figure 23 depicts the geometric interpretation when handling Equations (5.9)-(5.10).

5.3 Results, comparisons and evaluation

We begin this section showing the usefulness of our methodology in a practical application devoted to video data visualization. The flexibility to generate different layouts by only tuning the constraints in the optimization is illustrated in several examples. Finally, we show the effectiveness of MIOLA by comparing it against state-of-art overlap removal techniques

Figure 23 – Geometric interpretation of relaxation constants. Fig.(a) shows the input layout while Fig.(b) depicts the optimization result after setting upper bound relaxation (< 0) in both directions. Fig.(c)-(d) show the maximum distance tolerance when violating the orthogonal order constraints and imposing a displacement between boxes, respectively (in x-direction). Fig.(e)-(f) yield the same previous conclusion w.r.t. non-overlap constraints but obtained by handling y-direction relaxation.



(a) Input layout.

(b) Reducing upper bounds.

(c) Violating orthogonal order.

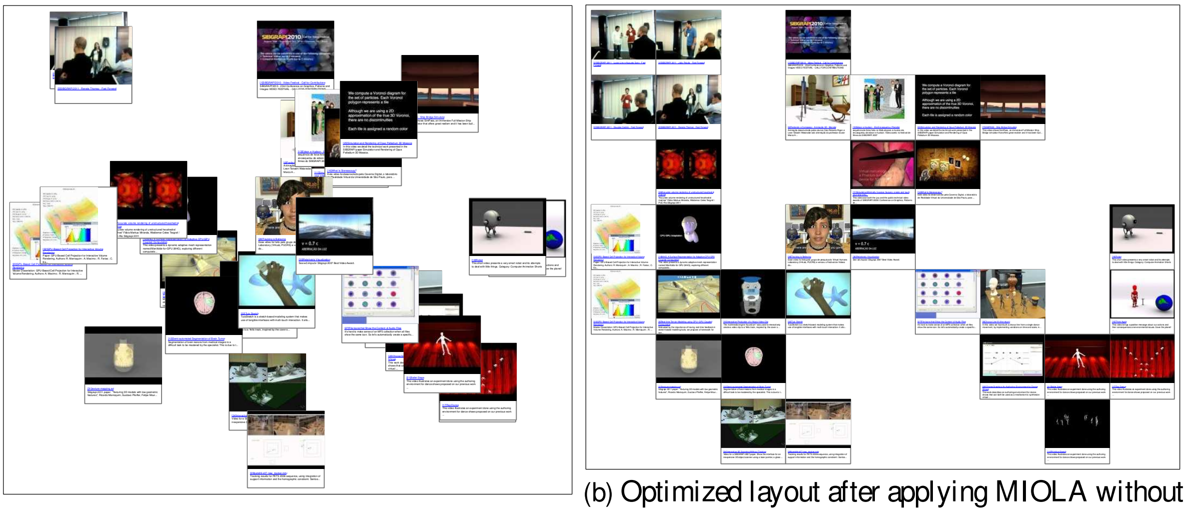
(d) Increasing orthogonal order.

(e) Increasing non-overlap distance.

(f) Violating non-overlap distance.

Source: Gomez-Nieto et al. (2013).

Figure 24 – Result of our method when applying on a content layout taken from Youtube's search engine.



(a) Input Layout.

(b) Optimized layout after applying MIOLA without relaxation constants.

Source: Gomez-Nieto et al. (2013).

typically employed in information visualization applications. We denote the pair of numbers $VS = ([-S_x, S_x], [-S_y, S_y]) := (S_x, S_y)$ as the lower and upper bound constants in (5.9). Orthogonal order as well as non-overlap relaxation constants in (5.10) are denoted by the numbers OO and NOL. The minimization problem (5.4) is solved by an extension of the Linear-Programming based Branch-and-Bound Algorithm (LEE; LEYFFER, 2012), available from the Gurobi library at <http://www.gurobi.com/>. Our code was implemented in MATLAB with support to C++ MEX routines and free optimization interface provided by Yin (2011).

5.3.1 Applications

Video Snippet Visualization Figure 24 shows the layout resulting from MIOLA applied to video snippet visualization. Figure 24a depicts the initial layout just after projecting video data obtained from Youtube, using the word "SIBGRAPI" as query. The projection is generated by applying the LSP multidimensional projection technique (PAULOVICH et al., 2008), measuring the similarity between video snippets through the cosine metric applied to feature vectors extracted from the textual data associated to the videos. More precisely, feature vectors are generated by extracting a bag-of-words (SALTON; WONG; YANG, 1975) from the text associated to each video. Figure 24b shows the resulting layout after optimization. Notice that SIBGRAPI's Fast Forward videos are nicely grouped on top left while videos related to volume rendering, modeling and graphics are arranged on middle/bottom left. Top right snippets in Figure 24b are mainly related to videos presented in SIBGRAPI's Video Festival while bottom right videos are related to "dancing animation". The video visualization application clearly shows that MIOLA is able to arrange the video snippets in the visual space avoiding overlaps while preserving the proximity relation among videos.

5.3.2 Flexibility to generate different layouts

Flexibility and Adaptability Figures 25a to 25e show the flexibility of MIOLA in producing different layouts by just modifying constraints in the optimization. Figure 25a shows the original layout while Figure 25b displays the resulting layout when no relaxation is used during optimization. Changing the constraints that define the lower and upper bounds of the visualization window give rise to a more compact layout, as shown in Figure 25c. We use colors and numbers to label the boxes, from which one can clearly see that MIOLA preserves neighborhoods quite nicely. An even more compact layout can be obtained by further squishing the visualization window and relaxing the orthogonality constraints, as depicted in Figure 25d. Figure 25e shows an extreme case where the visualization window is set to be one-dimensional, that is, the centroid of each box must be place in a horizontal line. Notice that even in this drastic situation the algorithm was able to arrange the square respecting neighborhoods.

Shrinking scattered layouts MIOLA can also be used for shrinking sparse layouts. Figure 26b shows the result of optimizing the spread layout depicted in Figure 26a. One can see that the optimization produces a pleasant layout in terms of compactness and organization. Moreover, neighborhood structures have also been nicely preserved.

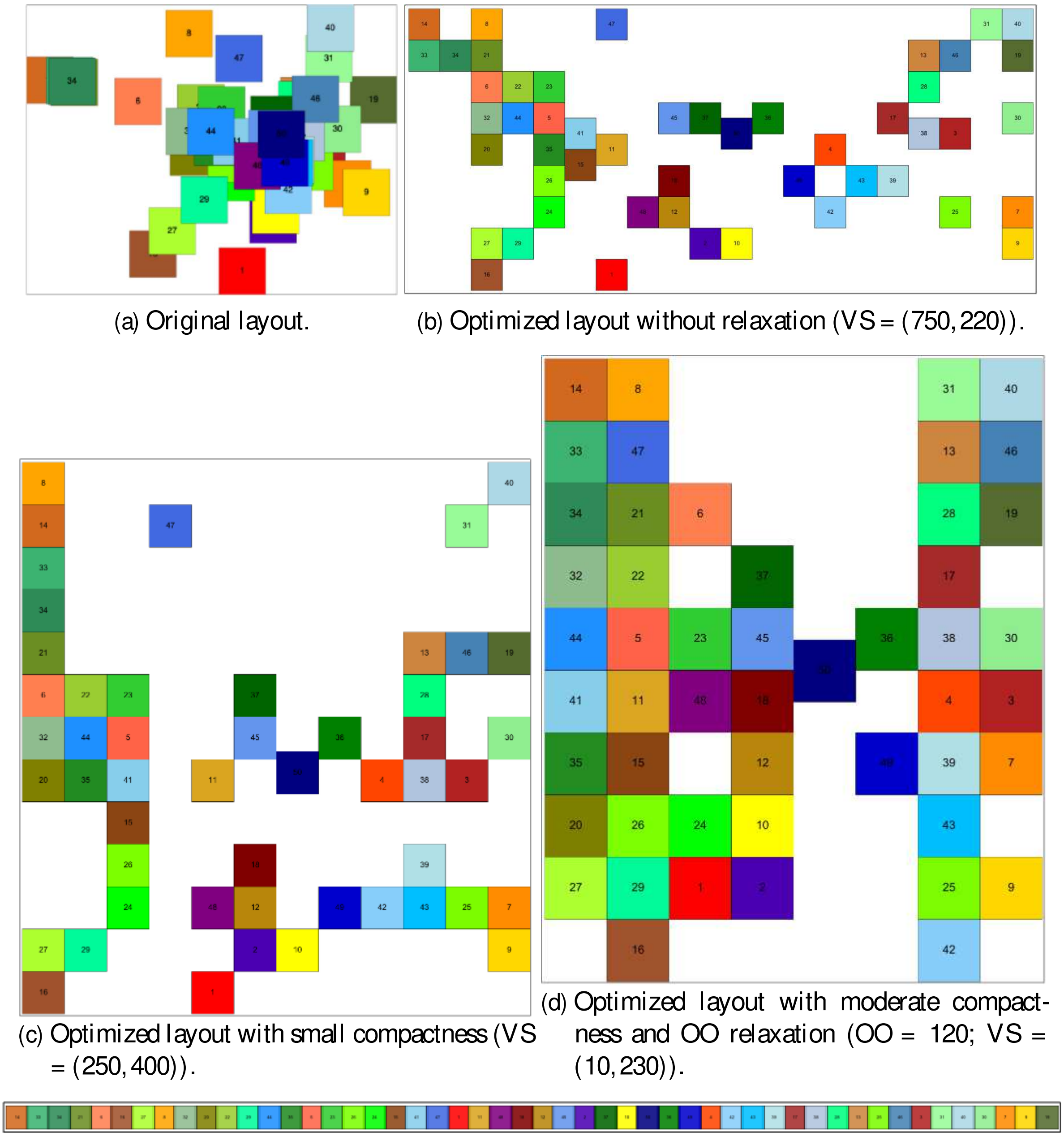
5.3.3 Comparisons against state-of-art methods

The effectiveness of MIOLA is assessed by quantitatively measuring its accuracy as to layout organization, neighborhood preserving and compactness against state-of-art methods. The quantitative measures are computed using the following metrics: Euclidean Distance (Section 2.4.1 referred here as E), Layout Similarity (Section 2.4.2 referred here as σ), Size Increase (Section 2.4.3 referred here as S), Orthogonal Ordering (Section 2.4.7 referred here as O) and Neighborhood Preservation (Section 2.4.5 referred here as k).

Such measures allow us to quantitatively compare our technique against the following well-established overlap removal methods: VPSC (DWYER; MARRIOTT; STUCKEY, 2006), PRISM (GANSNER; HU, 2009), Voronoi-based (DU; FABER; GUNZBURGER, 1999) and RWordle-C (STROBELT et al., 2012).

Figure 27 shows the resulting layout after applying MIOLA and the four techniques we compare against on five different datasets containing textual information collected from web search engines. The first three data sets, namely, “Scientific Visualization” (50 instances), “Sibgrapi” (32 instances) and “Purple”(100 instances), were generated using fixed-size square boxes while the last two datasets - “Wave” (64 instances) and “Batman” (50 instances) - were created taking rectangles with sizes given by the rank of the document (the rank is provided by the search engine). Larger rectangles have higher rank. One can note that the Voronoi-based method results in unstructured overlap-free layouts when compared to the other methods. PRISM preserves relevant parts of the initial shape, but with the disadvantage of excessive use of

Figure 25 – Flexibility and adaptability of the proposed formulation.



(a) Original layout.

(b) Optimized layout without relaxation (VS = (750, 220)).

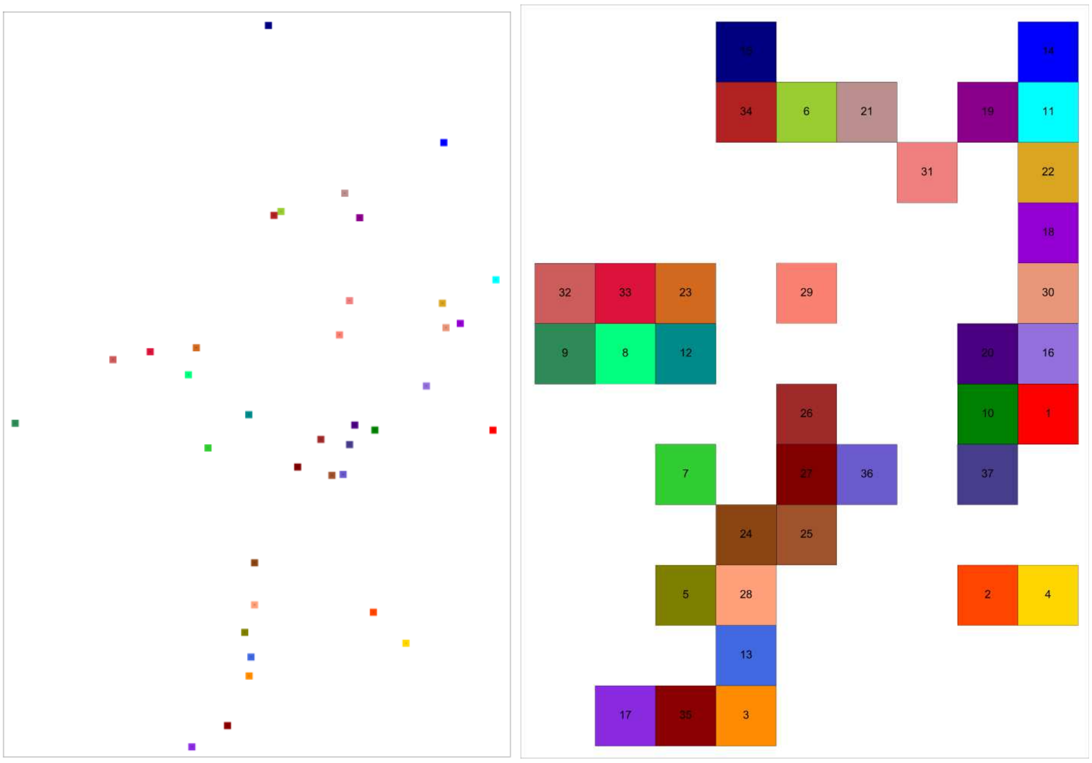
(c) Optimized layout with small compactness (VS = (250, 400)).

(d) Optimized layout with moderate compactness and OO relaxation (OO = 120; VS = (10, 230)).

(e) Optimized layout with hard compactness and NOL relaxation (NOL = 10; VS = (3800, 5)).

Source: Gomez-Nieto et al. (2013).

Figure 26 – Optimizing sparse layouts with MIOLA.



(a) Layout with scattered boxes.

(b) Result after layout adjustment.

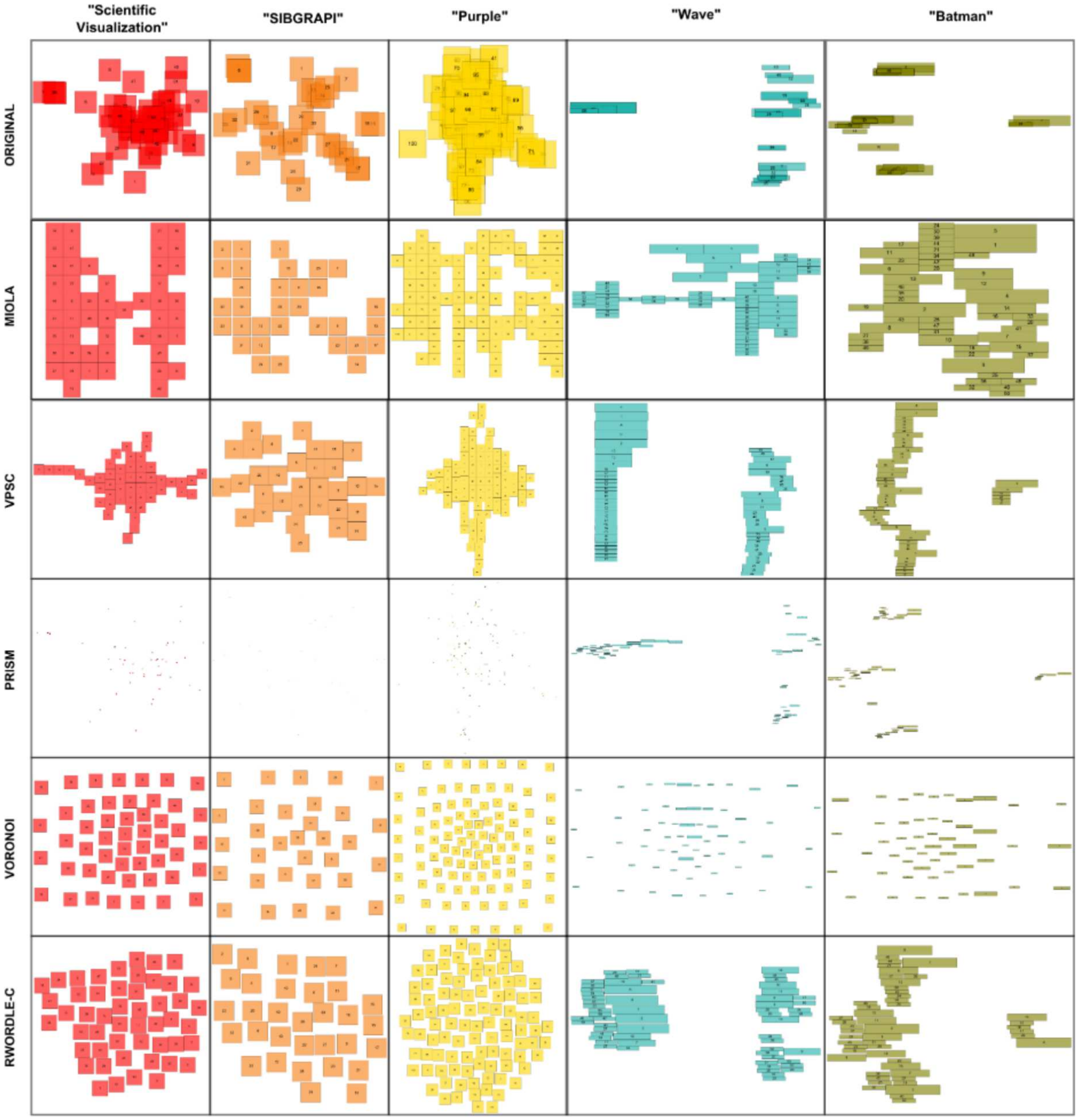
Source: Gomez-Nieto et al. (2013).

space, hampering data exploration. The layouts produced by VPSC are quite structured, helping readability, however, it is prone to stretch the layout vertically (see the "Wave" and "Batman" results). RWordle-C produced satisfactory layouts preserving, in some cases, clusters visually identified by the user (see the "Wave" result), but still leaving some expressive empty spaces when compared with MIOLA. Moreover, the orthogonal ordering is considerably affected by RWordle-C. In contrast, MIOLA produces very well-organized and compact layouts while still respecting the orthogonal order in a satisfactory way.

Figure 28 summarizes the quantitative results of the metrics described above. Notice that MIOLA produces more compact layouts measured by Euclidian distance and Size increase, being fairly stable as to other metrics.

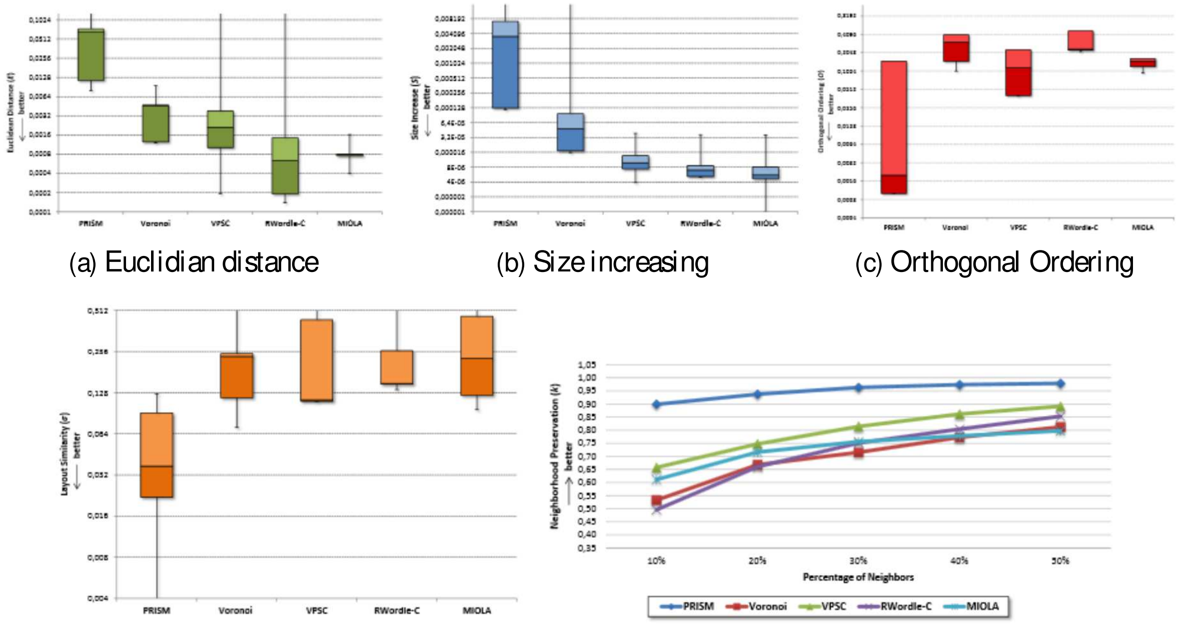
Figure 29 plots a map that considers all metrics simultaneously. Each overlap removal technique has been represented as a five-dimensional vector $(E_a, \sigma_a, O_a, S_a, k_a)$, where $E_a, \sigma_a, O_a, S_a, k_a$ are the average values of the metrics $\sigma_a - E_a$ and k-nearest neighbors computed for each technique, over all data sets from Figure 27. The blue points denoted by "Best", "Average", and "Worst" in 29 were created artificially as five-dimensional vectors containing the best, average and worst values taken from the results of all methods. We use the multidimensional projection method LAMP (JOIA et al., 2011) to project the five dimensional data to the visual

Figure 27 – Layouts produced by our approach (MIOLA), VPSC, PRISM, Voronoi-based, and RWordle-C for five distinct data sets.



Source: Gomez-Nieto et al. (2013).

Figure 28 – Comparing our approach, VPSC, PRISM, Voronoi-based, and RWordle-C considering the metrics E (a), σ (b), O (c), S (d), and k-nearest neighborhoods (e) (in log scale after normalization) regarding to Fig.27.



(a) Euclidian distance

(b) Size increasing

(c) Orthogonal Ordering

(d) Layout similarity

(e) k-neighborhood preservation

Source: Gomez-Nieto et al. (2013).

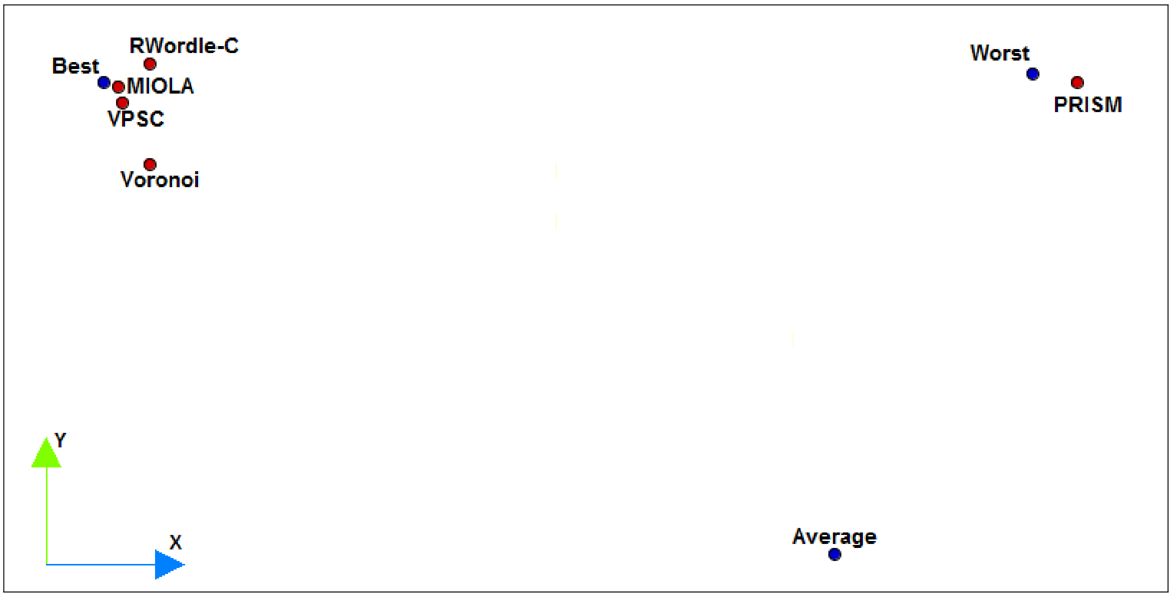
space. The technique mapped closest to "Best" is the one that, on average, has shown the best performance, relative to all metrics. Clearly, one can observe that MIOLA is the technique closer to the best result.

5.4 Discussion and limitations

Our technique displays layouts in such a way that keeps boxes disjointed while still preserving the similarity of the neighborhood structures. Comparisons presented in Section 5.3 clearly show the effectiveness of the proposed optimization method, surpassing, in terms of accuracy, existing methods. As shown in Figure 25, MIOLA turns out to be flexible and capable of producing a variety of layouts, a characteristic not present in any other state-of-art method. The snippet video application depicted in Figure 24 indicates that MIOLA can be used to remove overlaps while keeping similar entities close to each other, thus preserving the similarity relations introduced during the multidimensional projection.

Despite good results and solid mathematical foundation, there are two aspects to be observed when using MIOLA. First, the visualization window as well as the number of boxes to be displayed simultaneously can affect the readability of the visualization. Easy to read layouts depend on the screen size. Finding out a good trade-off between clean/pleasant layouts and the relaxation parameters is another issue that deserves further investigation.

Figure 29 – Comparison of overlap removal techniques respecting the five metrics simultaneously. Red points represent the evaluated techniques while the blue points show the worst, average and best points artificially produced.



Source: Gomez-Nieto et al. (2013).

5.5 Conclusion

In this chapter we described a technique for solving the box overlap removal problem, called MIOLA. The evaluation we provided shows that the proposed method outperforms existing methods in terms of producing well structured layout. MIOLA turns out to be effective in practical application and quite flexible to generate different layouts by just tuning parameters, rending it a very attractive overlap removal approach.

CHAPTER

6

DEALING WITH MULTIPLE REQUIREMENTS

6.1 Introduction

In this chapter 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.

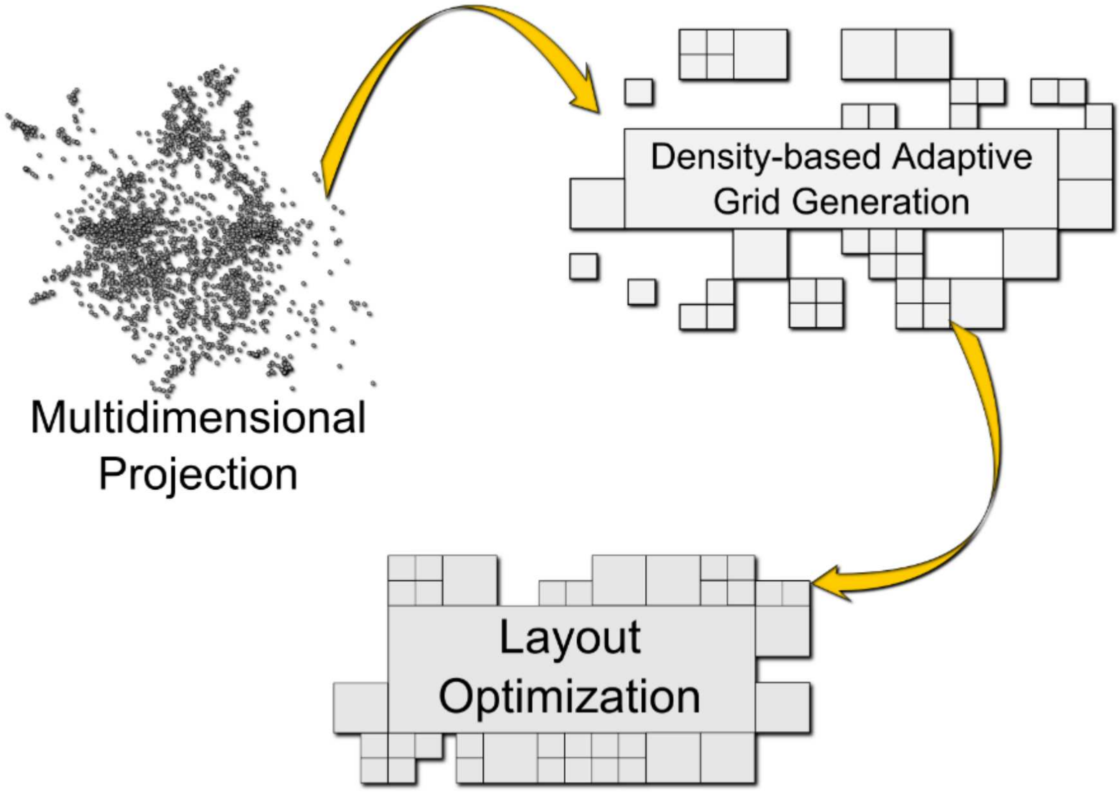
The proposed optimization scheme arranges geometric entities (boxes) with varying sizes so as to avoid overlaps while preserving the neighborhood structure of the underlying data (semantics). The area of each geometric primitive is also included in the optimization process to ensure that the display 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. A comprehensive set of quantitative comparisons against existing geometry-based layouts shows the effectiveness of our approach. The usefulness of our methodology in visualizing text, image, and video data sets is also confirmed in several practical applications.

6.2 The method

Figure 30 illustrates the proposed methodology 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 (PAULOVICH et al., 2008), which preserves distances nicely during the mapping

Figure 30 – Pipeline used comprises three main steps: (i) data is mapped into visual space using a multidimensional projection technique, (ii) adaptive grid is built from projected points and (iii) an optimization algorithm is performed to reach an optimal area usage.


Multidimensional Projection
Density-based Adaptive Grid Generation
Layout Optimization

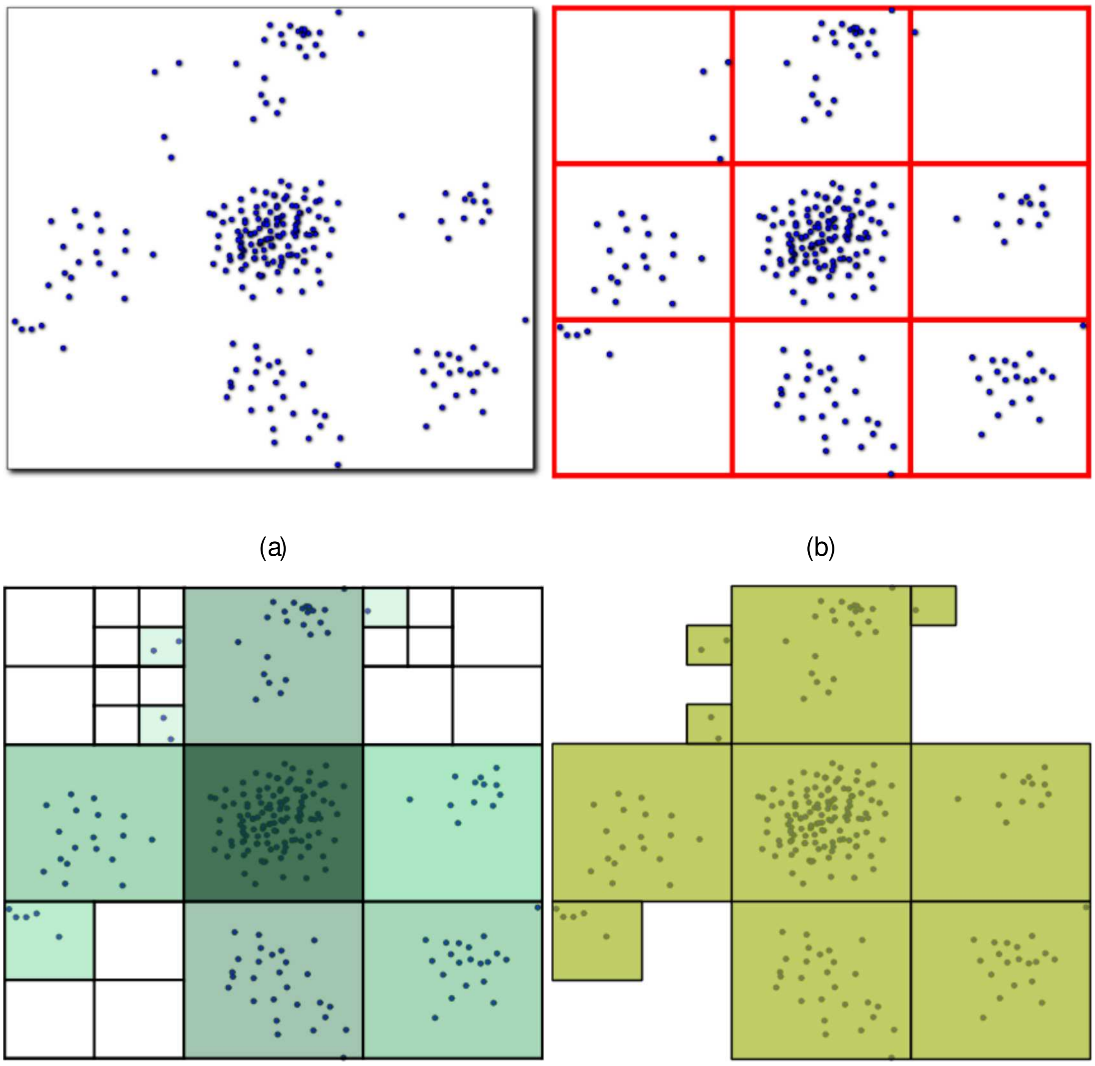
Source: Gomez-Nieto et al. (2016).

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 (see Figure 30 top right). 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 (see Figure 30 bottom).

Figure 31 – Density based refinement process. (a) An initial regular grid is defined in the visual space. Such an initial grid provides the coarser refinement level (first level, as depicted in (b)). (c) An adaptive grid is built from density information. Denser areas are represented by larger cells. (d) Active (not empty) cells are then ready to be optimized in the next step of the proposed framework.



Source: Gomez-Nieto et al. (2016).

The rationale here is to build layouts as readable as possible while still dragging users attention to denser regions that should be further explored.

The following subsections detail the second and third step of our pipeline, which correspond to the major technical contributions of our method.

6.2.1 Adaptive grid generation

Let $P = \{p_1, p_2, \ldots, p_q\} \subset \mathbb{R}^2$ be the projection of a set of instances into the 2D visual space and G be a $n \times m$ regular grid which discretizes the bounding box of P. The number of cells in each orthogonal direction is defined such that each grid cell g_{ij} has a square shape, that is, given the number of subdivision in one direction, for instance m, the number of subdivision in the other direction is $n = \lceil \frac{H}{W} m \rceil$ (the boundary of the bounding box can be displaced to accommodate the number of cells), where H and W account for the height and width of the bounding box of P , respectively.

The regular mesh G is the coarser grid level for the adaptive process. The refinement is driven by density information computed from P in each grid cell. The density can be estimated in different ways, for instance by integrating a kernel density estimator over each grid cell. As our application does not demand highly accurate density estimation, we opt to approximate the density in each cell by simply counting the number of points p_i contained in the cell.

Given the density calculation, the refinement process is carried out as follows: let d_{ij} be the density of the grid cell g_{ij} and $d_{max} = \max\{d_{ij}\}$ be the largest density value in G. A grid cell g_{ij} undergoes one level of refinement if $d_{ij} \leq 0.5d_{max}$, two levels of refinement if $d_{ij} \leq 0.25d_{max}$, three levels if $d_{ij} \leq 0.125d_{max}$, and so on. Figure 31 illustrates the refinement process with three levels.

Although simple, the grid generation scheme described above has two meaningful properties. First, dense regions in the visual space can easily be identified from the larger cells, pointing out where users should nail down their exploration. Second, the proposed refinement scheme allows for resizing all grid cells by controlling only one parameter. In other words, if one wants to scale all cells preserving their relative size, the only parameter to be tuned is the length δ of coarser cells. In fact, the length of the cells in each refinement level is given by $(1/2^k)\delta$, where $k = 0, 1, \cdots$ is the level of refinement. This last property will be exploited during optimization to find the value of δ that results in the best use of display area, as detailed below.

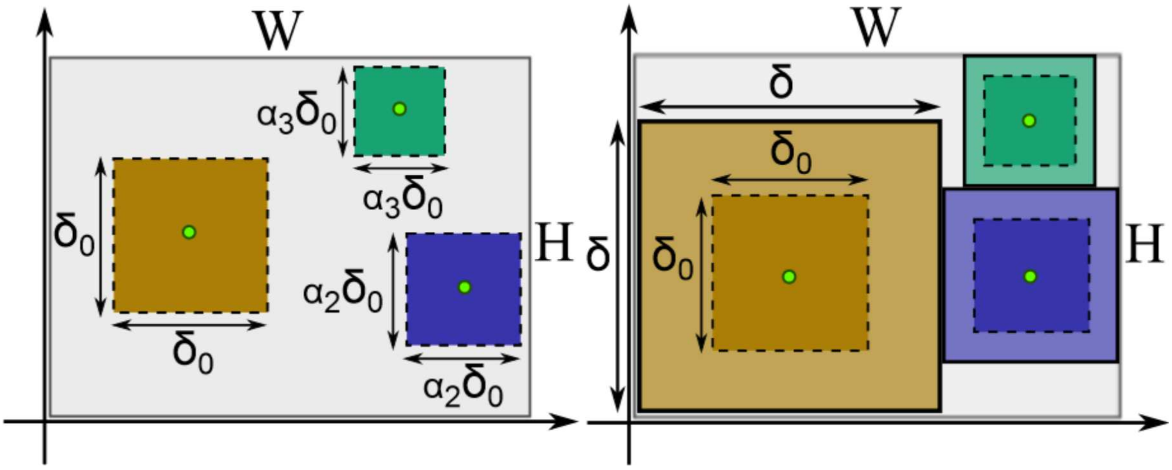
6.2.2 Optimization

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 $g_i = (x_i, y_i, w_i) \in R^3$, 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. As described in Section 6.2.1, $w_i = \alpha_i \delta$, where $\alpha_i = 1/2^k$, k corresponding to the level of refinement of g_i. Hence, δ is a parameter to be optimized (initially set as the length δ_0 of the coarser cell) from which the size of each cell is derived, as Figure 32 shows.

6.2.2.1 Rearranging cells with area resizing

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 (LEUNG et al., 1990) present in different contexts

Figure 32 – (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 $\delta > \delta_0$.



(a) (b)

Source: Gomez-Nieto et al. (2016).

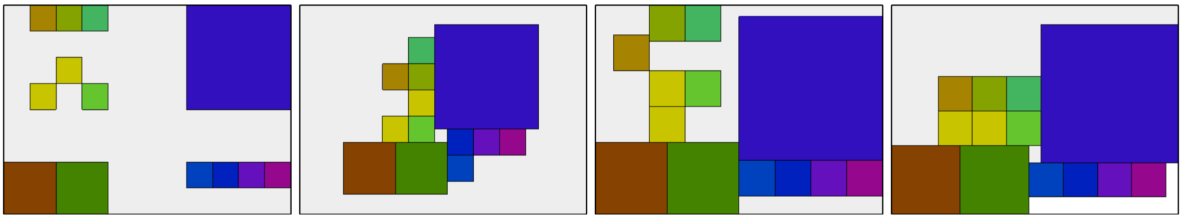
such as cutting optimization and rectangle packing (JANSEN; PRADEL, 2009). In order to get an approximate solution, we formulate the problem as a computationally tractable quadratic optimization problem. The optimization is formulated as in Equation (6.1) below:

$$
\begin{aligned}
&\text{minimize } E(z) = E_{comp}(z) + E_{rezise}(z),\\
&\text{subject to } \; Az \leq b, \quad z = [x \;\; y \;\; r \;\; \delta]^{\top},\\
&\;\; x = (x_1, x_2, \ldots, x_N)^{\top} \in \mathbb{R}^N\\
&\;\; y = (y_1, y_2, \ldots, y_N)^{\top} \in \mathbb{R}^N\\
&\;\; r = (r_{12}, \ldots, r_{1N}, r_{23}, \ldots, r_{2N}, \ldots, r_{N-1N})^{\top}, r_{ij} \in \{0, 1\}\\
&\;\; \delta_0 \leq \delta \leq \min(W, H),
\end{aligned}
\tag{6.1}
$$

where z is the sought solution; y and x correspond to the coordinates of the centroids of the cells; δ is the scaling factor; A and b hold the constraints imposed on the optimization problem. The unknowns r_{ij} are control variables used to properly avoid overlaps. The energy components $E_{comp}(z)$ and $E_{resize}(z)$ 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, as Figure 33 illustrates.

Compactness Energy Term The first term in the energy functional, E_{comp}, controls the relative position among cells and its goal is to keep the layout compact. The energy E_{comp} is a quadratic function that simply takes into account the centers of the boxes, as shown in

Figure 33 – Minimizing each term of the energy functional E in an illustrative layout. (a) Original layout, (b) E_{comp} only, (c) E_{resize} only, (d) the proposed energy functional $E = E_{comp} + E_{resize}$.



(a) (b) (c) (d)

Source: Gomez-Nieto et al. (2016).

Equation (6.2):

$$E_{comp}(z) = C \sum_{(i,j)} (x_i - x_j)^2 + (y_i - y_j)^2, \tag{6.2}$$

where $(i, j) = (j, i)$ represents the $\frac{N(N-1)}{2}$ pairs $(g_i, g_j), i, j, \in \{1, ..., N\}$ and $C = 1/ \ \min(W, H) \cdot \frac{N(N-1)}{2}$ is a normalization factor. In less mathematical terms, E_{comp} forces boxes to be close to each other and the normalization factor C ensures that the term E_{comp} contributes in the same amount as E_{resize} in the total energy E, thus enabling a good balance between compactness and area usage.

Area Usage Energy Term The energy term E_{resize} controls the amount boxes should be scaled to optimize the use of display area. Since the area of any cell depends only on the length of the largest cells, that is, the area of each cell g_i is given by $A_i = (\alpha_i \delta)^2$, $\alpha_i = 1/2^k$, where k is the refinement level of g_i, we can define the resize energy as a quadratic function as follows:

$$E_{resize}(z) = (\delta - \min(W, H))^2, \tag{6.3}$$

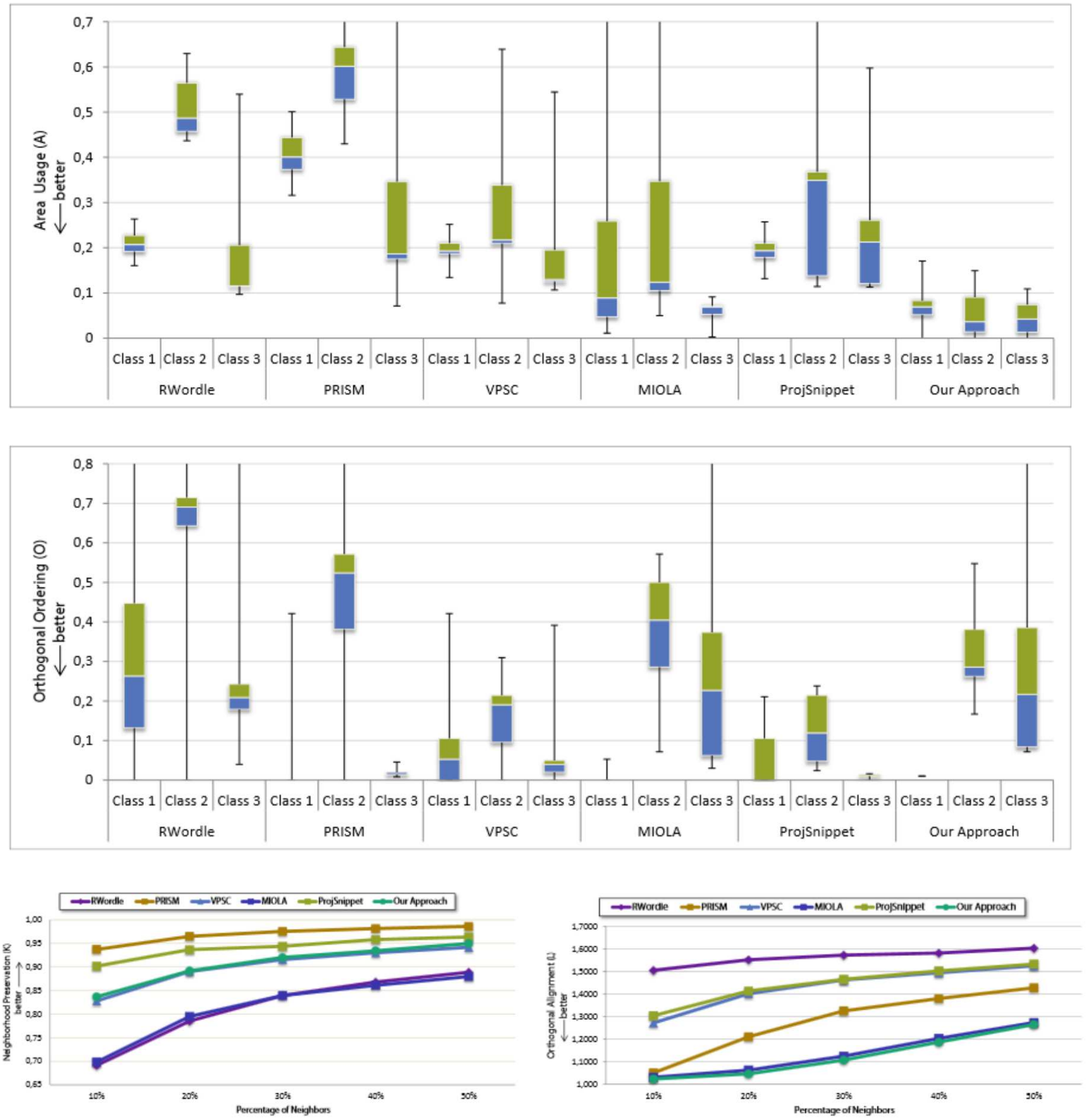
where $\min(W, H)$ is the minimum between the width and height of the display area (see Fig. 33c).

It is easy to see that if no constraints are imposed to the unknowns, the minimum of E_{comp} is reached when all cells have the same center and the minimum of E_{resize} takes place when the length of the larger cell is equal to $\min(W, H)$. Therefore, without constraints the optimization process will stack cells on over the other. To avoid such unsuitable output, we must settle constraints as discussed next.

6.2.2.2 Optimization constraints

As described in Equation (6.1), constraints are gathered in matrix A and vector b and they are imposed to properly control overlaps and the relative position of cells. Given the initial position (x_i, y_i) and length $\alpha_i \delta$ of each cell, the relative order (also called orthogonal order) of the boxes is given by:

Figure 34 – Comparison against overlap removal techniques for the three initial classes of experiments.



Source: Gomez-Nieto et al. (2016).

$$\begin{array}{l} x_{p_1} \leq x_{p_2} \leq \ldots \leq x_{p_n} \Rightarrow x_{p_i} - x_{p_{i+1}} \leq 0 \\ y_{q_1} \leq y_{q_2} \leq \ldots \leq y_{q_n} \Rightarrow y_{q_i} - y_{q_{i+1}} \leq 0 \end{array} \tag{6.4}$$

where $p, q : \{1, 2, ..., N\} \rightarrow \{1, 2, ..., N\}$ are permutations of indices generated by sorting the coordinates x_i and y_i.

Inequalities (6.4) allows for preserving the relative order of the cells, but it does not account for overlap. Overlaps can be handled by forcing non-overlap conditions as follows:

$$|x_j - x_i| \geq \frac{(\alpha_i + \alpha_j)}{2}\delta \quad \text{or} \quad |y_j - y_i| \geq \frac{(\alpha_i + \alpha_j)}{2}\delta, \tag{6.5}$$

From Equation (6.4) we have that if $x_i \leq x_j$ and $y_i \leq y_j$ then $|x_j - x_i| = x_j - x_i$ and $|y_j - y_i| = y_j - y_i$, which give rise to the following linear system of inequality:

$$\begin{array}{c} x_i - x_j \leq -\frac{(\alpha_i + \alpha_j)}{2}\delta, \\ \text{or} \\ y_i - y_j \leq -\frac{(\alpha_i + \alpha_j)}{2}\delta, \end{array} \tag{6.6}$$

As proposed in (GOMEZ-NIETO et al., 2013), the or condition in (6.6) can be handle by binary variables $r_{ij} \in \{0, 1\}$ such that:

$$x_i - x_j \leq \alpha_{ij}\delta + Mr_{ij} \Leftrightarrow y_i - y_j \leq \alpha_{ij}\delta + M(1 - r_{ij}), \tag{6.7}$$

where $\alpha_{ij} = -\frac{(\alpha_i + \alpha_j)}{2}$ and M is a very large constant. The role of r_{ij} is to ensure that if one of the inequalities in Equation (6.6) holds, its counterpart is set aside. For instance, if $x_i - x_j \leq \alpha_{ij}\delta$, then r_{ij} is equal to zero, thus $y_i - y_j \leq \alpha_{ij}\delta + M(1 - r_{ij})$ is naturally satisfied for any values of y. The linear inequalities, r_{ij} unknowns, and δ make up the linear system $[A|b]$. The left and right bounds for the x coordinate of the center of the cells must also be imposed in $[A|b]$, that is,

$$0 \leq x_i - \frac{\alpha_i}{2}\delta \quad \text{and} \quad x_i + \frac{\alpha_i}{2}\delta < W, \; i = 1, 2, ..., N, \tag{6.8}$$

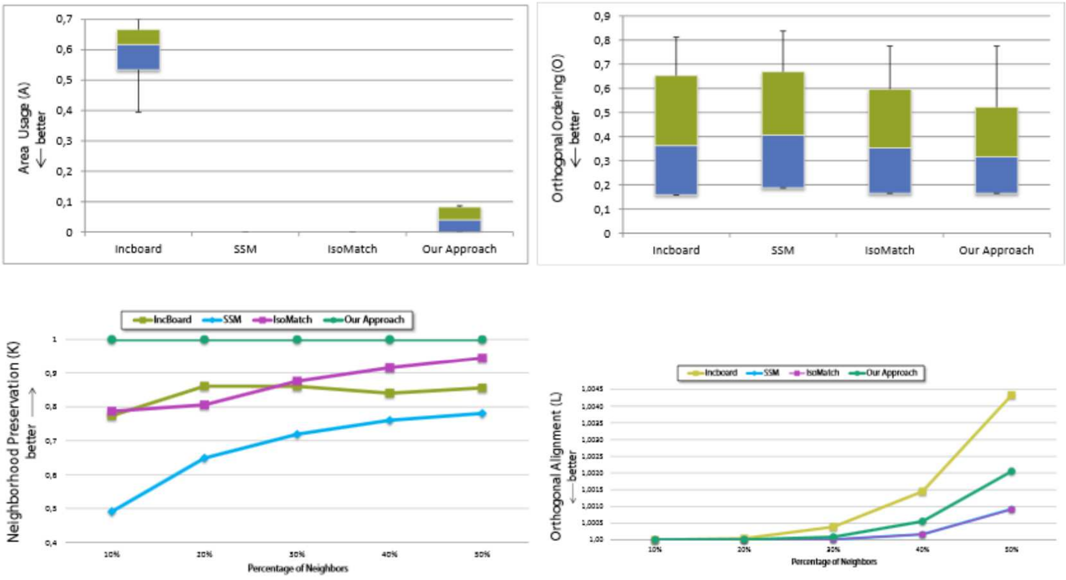
Similar inequalities are set to limit the upper and lower bounds of the y coordinates.

Constraints Relaxation The optimization problem described in Equation (6.1) provides a flexible and simple mechanism to relax constraints and generate distinct layout arrangements. This is performed by adding relaxation parameters (scalars) in the inequalities to allow for some degree of overlap, orthogonal disorder and so on.

6.2.2.3 Computational aspects

The formulation (6.1) is a Mixed Integer Quadratic Programming (MIQP) Problem, which is minimized by computing an extension of the so-called Branch-and-Bound optimization scheme, where the original MIQP problem is converted to a Linear Programming Problem in order to be properly handled and solved (LEE; LEYFFER, 2012).

Figure 35 – Comparison with visual board techniques for the last class (taken from Fig. 37).



Source: Gomez-Nieto et al. (2016).

In terms of usage, our code was implemented using the solvers and optimization routines provided by Gurobi Optimization Package, which is available at <http://www.gurobi.com/>.

6.3 Results and comparisons

The performance of the proposed optimization scheme is assessed through a set of comparisons against well established geometry based layout construction techniques. More precisely, we employ four distinct metrics to quantitatively measure the quality of layouts produced by two distinct classes of techniques, namely: overlap removal methods and visual board techniques. Before presenting the comparison, we briefly describe the metrics as well as the data sets used in our tests.

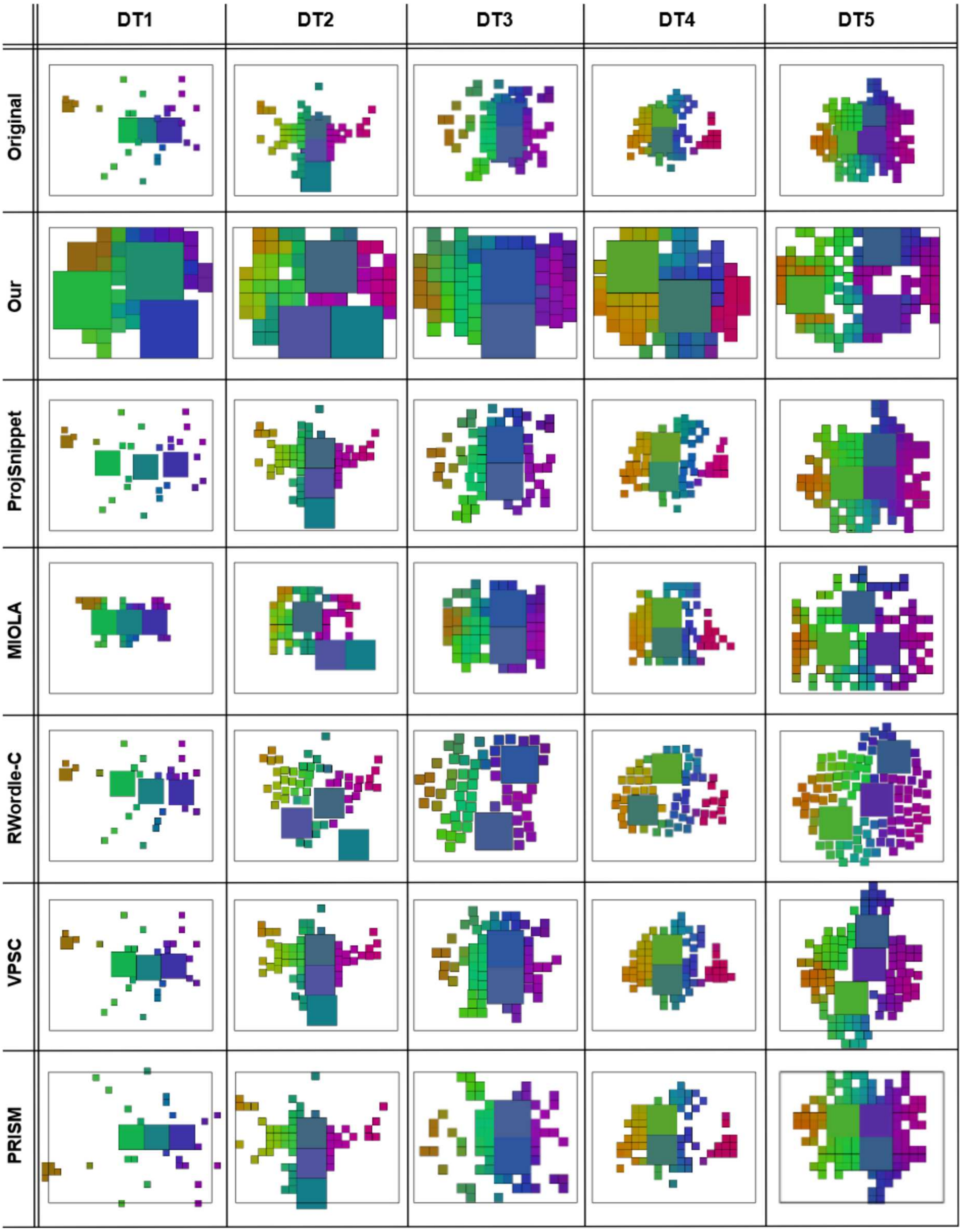
6.3.1 Metrics and datasets

The quantitative measures are computed using the following metrics: Orthogonal Ordering (Section 2.4.7 referred here as O), Neighborhood Preservation (Section 2.4.5 referred here as K), Area Usage (Section 2.4.9 referred here as A) and Orthogonal Alignment (Section 2.4.8).

Metrics above have been chosen because they measure important properties a semantic layout must hold, namely, similarity preserving, compactness, and orthogonal alignment.

Data sets and Test Settings We divided the experiments in four groups: the first three groups of tests assess overlap removal methods while the last group evaluates the performance of our optimization scheme against visual board techniques.

Figure 36 – Layouts produced by our approach, ProjSnippet, MIOLA, RWordle-C, VPSC and PRISM in 5 datasets.



Source: Gomez-Nieto et al. (2016).

In the first test we randomly draw points inside a rectangular two-dimensional domain, which represents the display area. The rectangle is discretized as a regular grid with dimensions 1600 × 1200. Cells of the grid that do not contain any point are discarded. The remaining cells define the boxes to be arranged in the layout. As no overlap exists at this point, we enforce the cells to overlap by uniformly increasing their area in 10%. Forcing overlap is mandatory since most overlap removal methods can not start up from an overlap free arrangement. Notice that this is not the case for our approach, which also operates on overlap free layouts in order to optimize area usage and structure. Fifteen configurations, with fifty random points each, make up the first group of tests (called class 1 in Figure 34). The second experiment is similar to the first one, except we use a three level adaptive grid to generate the geometric primitives (class 2 in Figure 34).

In the third experiment (class 3 in Figure 34) we use real data set containing high-dimensional instances which are mapped to the visual space using the LSP multidimensional projection scheme. The layout is built from the projected points using a three level adaptive grid. Five data sets, taken from a public repository[1], have been employed in the third experiment. One data set contains a collection of snippets retrieved from a web search using "Scientific Visualization" as query expression; two data sets contain news from BBC, CNN, and REUTERS; the fourth data set corresponds to abstracts from IEEE InfoVis 2004 conference; and the last data set contains metadata from a scientific paper collection. Bag-of-words were extracted from textual content and used to define the similarity between instances. Those distances are used as input to the LSP projection method (PAULOVICH et al., 2008), which maps instances to the visual space.

Finally, in the fourth group of tests we have randomly generated squares in a 2D square domain, as depicted in Figure 37. Those layouts are used as input to existing visual board techniques.

6.3.2 Comparisons

The metrics describe above are used to assess the effectiveness of our approach when compared against well known overlap removal techniques. Precisely, VPSC (DWYER; MARRIOTT; STUCKEY, 2006), PRISM (GANSNER; HU, 2009), RWordle-C (STROBELT et al., 2012), MIOLA (GOMEZ-NIETO et al., 2013), and ProjSnippet (GOMEZ-NIETO et al., 2014) are used as basis for comparisons. These methods have been chosen either because they are widely used or due to their good performance reported in the literature. We also compare our approach against three visual board techniques, namely, Incboard (PINHO; OLIVEIRA; LOPES, 2010), SSM (STRONG; GONG, 2014) and IsoMatch (FRIED et al., 2015).

Figure 34 shows quantitative results obtained from the metrics described above. Notice that our approach clearly outperforms most techniques as to area usage (top row) and orthogonal alignment (bottom row right). Regarding orthogonal ordering (middle row), our approach turns out to be competitive, performing better for class 1. In terms of neighborhood preservation (bottom row left), our approach also performed well, being surpassed only by ProjSnippet

[1] Available at <http://infoserver.lcad.icmc.usp.br/infovis2/DataSets>

and PRISM, which are known to preserve neighborhoods well. Figure 35 brings quantitative comparison against visual board techniques. Notice that the proposed method outperforms IncBoard and behaves nicely even when compared to SSM and IsoMatch, which have optimal area usage (notice the vanish box plots) and cell alignment.

Figures 36 and 37 depict 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.

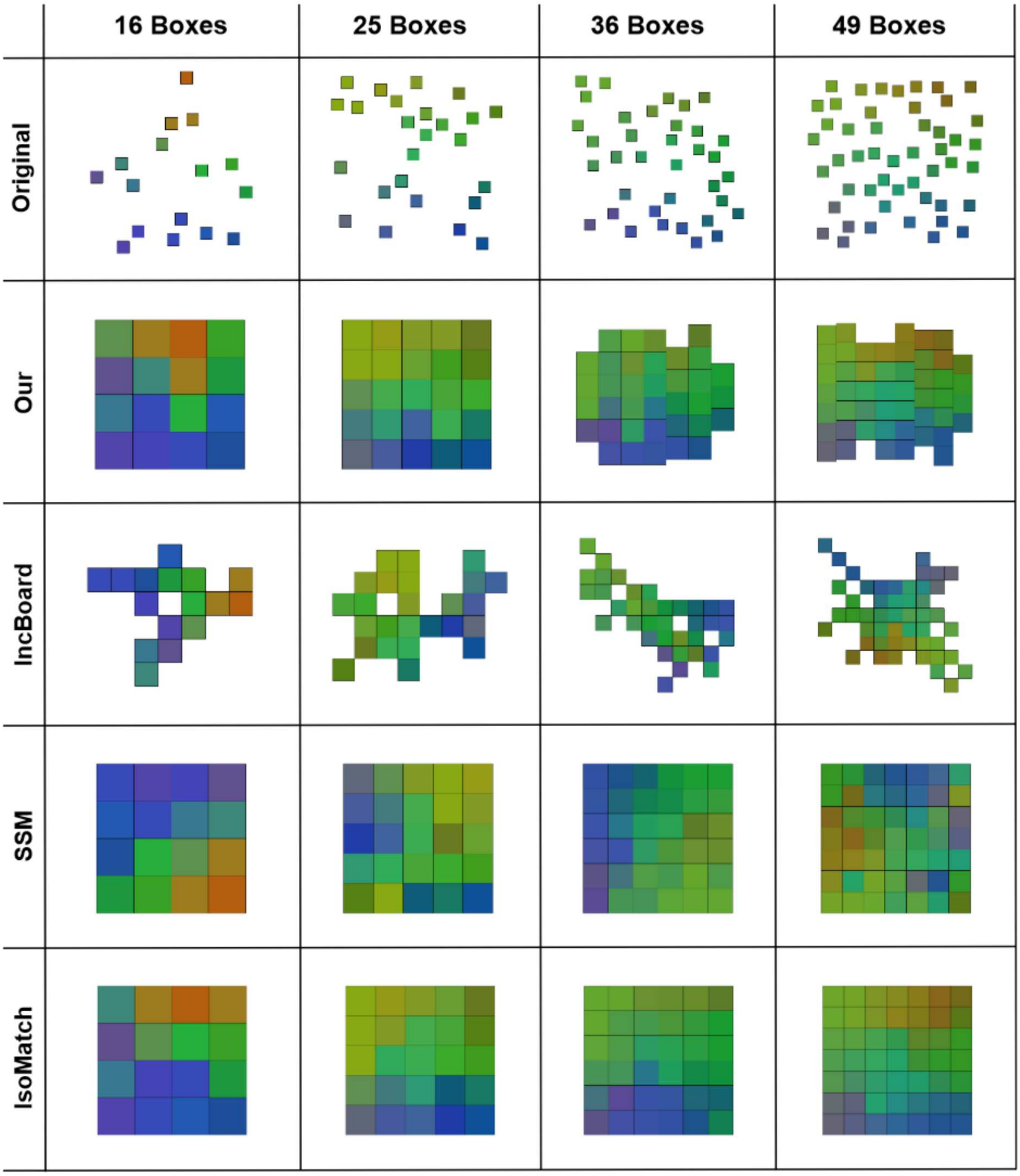
6.4 Applications

In this section we present three different applications of our technique, namely, image gallery construction, textual documents analysis, and video data set visualization. In those applications, effective data exploration is enabled if requirements such as object size, semantic and overlap-free arrangements, scalability, and optimal area usage are combined in an optimal manner, showing the relevance of our methodology to applications as the ones presented in this section.

The image gallery application, depicted in Figure 38, aims at visualizing image data sets such that highly similar images, which tend to be projected close to each other in the visual space, are summarized in large icons, while images with discriminative feature are represented in small icons. To build the image gallery we extract 96 features from color attribute. More specifically, each image is split in 16 non-overlapping regular regions covering the image from which the first and second statistical moments for each R,G,B channel is extracted. Feature vectors representing the images are projected in the visual space using the Least Square Projection (LSP) (PAULOVICH et al., 2008) method. A three level adaptive grid is constructed and the proposed layout optimization scheme is triggered in order to arrange the cells in a structured overlap free layout. Large grid cells containing several similar images give rise to iconic representations that summarizes the underlying images. Such iconic representation is built by blending a randomly chosen subset of the underlying images using five levels of Laplacian pyramid (DO; VETTERLI, 2003). In the example shown in Figure 38, cells containing less then four images are represented by a single representative image, cells containing four to eight images gives rise to a four blended image icon, and icon with nine blended images is used to represent cells with nine or more images. Images used in this application were obtained from (FEI-FEI; FERGUS; PERONA, 2007).

The second application concerns textual document visualization. Following the conventional tf-idf bag-of-words construction (SALTON, 1991) and stemming (LUHN, 1958), one can yield feature vectors from which our methodology can straightly be applied. In the application illustrated in Figure 39, we visualize a collection of 515 related papers published in the IEEE

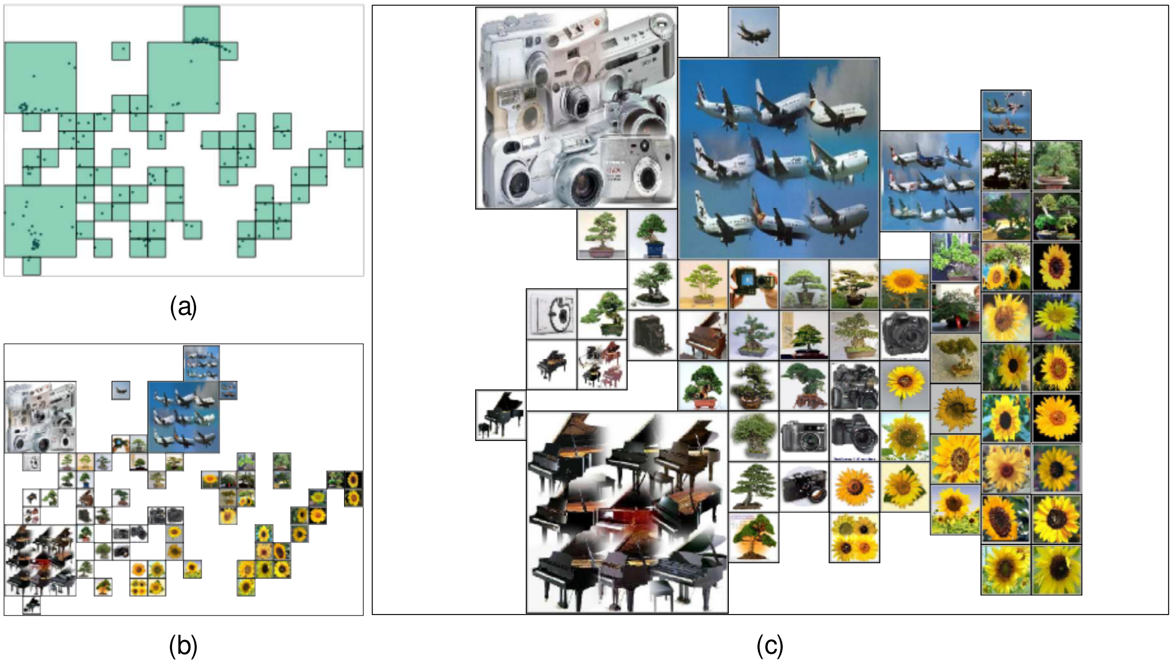
Figure 37 – Qualitative comparison of the proposed optimization scheme against visual board techniques Incboard (PINHO; OLIVEIRA; LOPES, 2010), SSM (STRONG; GONG, 2014) when taking the best result from SSM w.r.t. their objective function after 1000 executions and IsoMatch (FRIED et al., 2015)
.



Source: Gomez-Nieto et al. (2016).

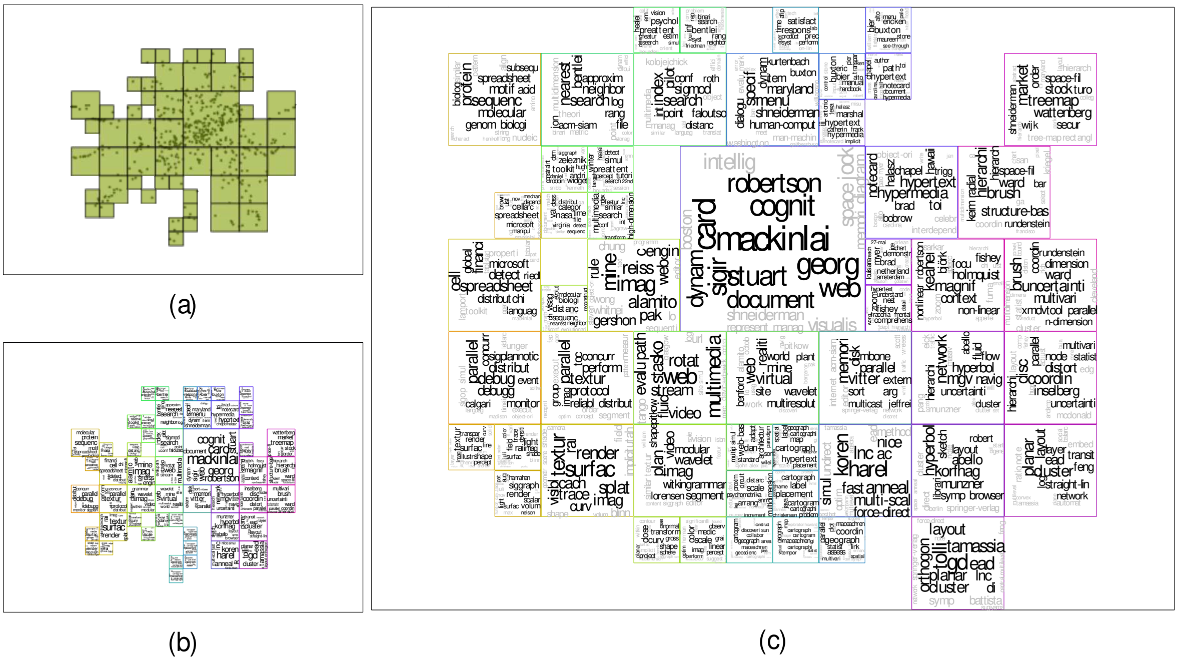
VisWeek Conference 2004. Each cell is textured with a word cloud built from keywords extracted from the documents contained in the cell while font size and opacity is used to highlight the most relevant keywords whose ranking is computed using the method described in (PAULOVICH et al., 2012). Notice that the layout is easy to read, mainly due to the effective use of display area. Moreover, it is not difficult to note that the grid-like structure makes the visual identification of

Figure 38 – Image gallery application: (a) an adaptive grid is built from projected images; (b) an initial iconic representation is generated; (c) the layout is optimize to make an optimal use of area while preserving the neighborhood of the grid cells.



Source: Gomez-Nieto et al. (2016).

Figure 39 – Visualizing a dataset containing 515 documents using word clouds.(a) each document is represented by a point embedded in 2D space where adaptive grid is imposed, (b) a word cloud based on term-frequency of documents contained in each cell of grid is built.(c) shows the optimized layout using the proposed method. Notice the size increasing that allow us to increase the number of words by each cell (colored in lightgray).



Source: Gomez-Nieto et al. (2016).

similar documents an easier task.

Our last application regards video visualization, as illustrated in Figure 40. We build a collection of 300 videos 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. As in the previous application, font sizes and priority-based opacity are used to highlight the importance of each keyword, which are ranked as in (PAULOVICH et al., 2012).

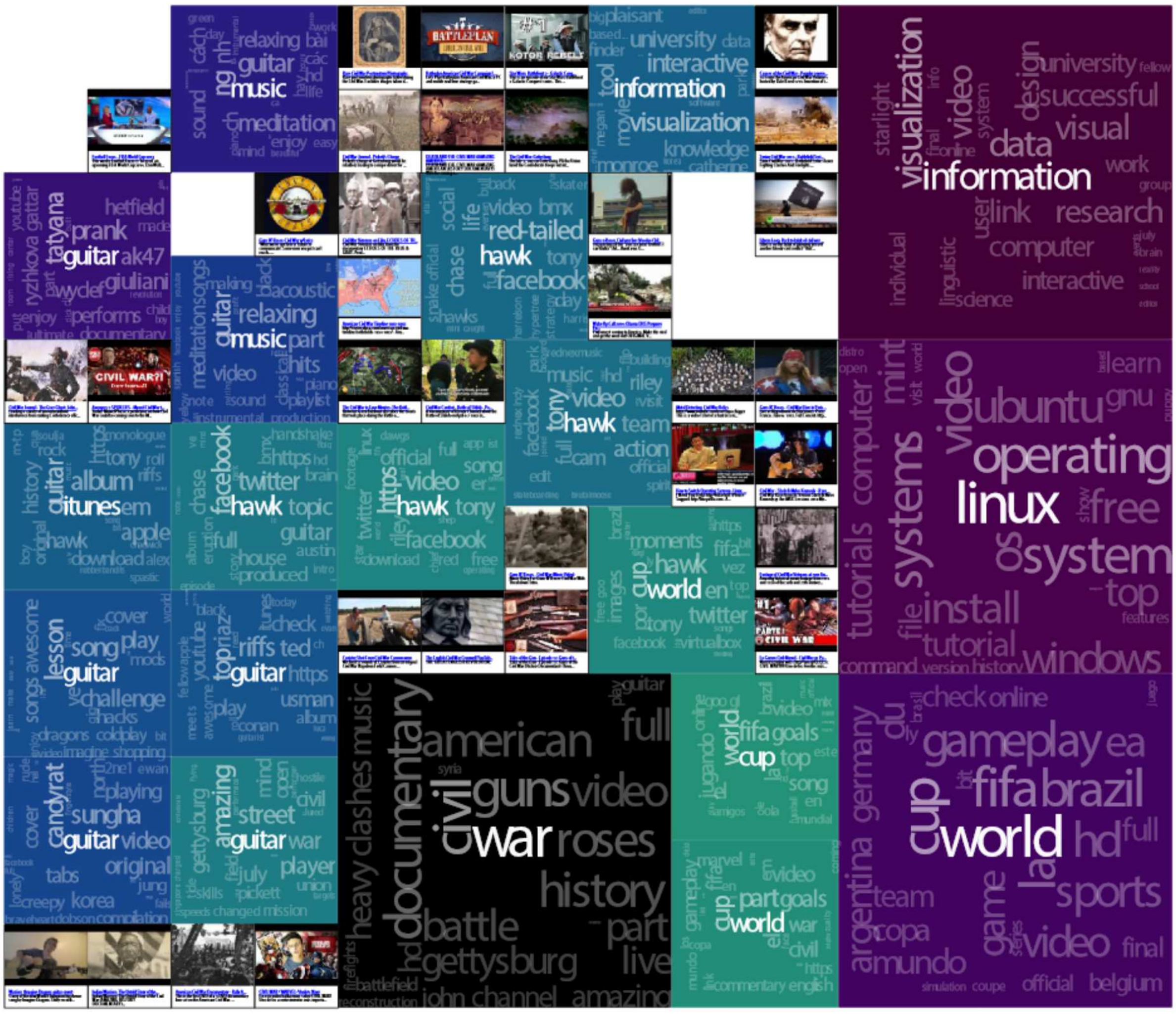
Notice how our approach was able to semantically organize the layout according to the distinct topics, keeping videos related to “guitar” grouped on the left part of the layout, “linux” and “information visualization” on the top right, “hawk” in the center, “fifa world cup” on the bottom right, and “civil war” centered on the bottom. Interestingly, the smaller cells in the layout contain mainly videos about civil war and music videos related to the song “Civil War” by Guns N’ Roses. Moreover, besides being well structured and semantically organized, the layout resulting from our methodology is compact and easy to read due to its effective use of display area. One interesting fact of all generated layouts is the high level of interactivity for the exploration task i.e. users are allowed to choose a cell summarizing some content (from all levels except the last one) and then further explore that cell in more detail by recursively applying the whole layout construction process only within that cell.

6.5 Discussion and limitations

Quantitative and qualitative comparisons presented in Section 6.3 attest the quality of our layout arrangement mechanism. In fact, our approach clearly outperforms most existing techniques with respect to properties such as area usage, orthogonal ordering and layout alignment, in addition to perform well as to neighborhood preservation. Therefore, our approach is able to handle several requirement simultaneously, making it an attractive alternative for several applications. The applications described in Section 6.4 show the versatility of our approach in scenarios ranging from image gallery construction to word cloud based text visualization. Moreover, the layouts resulting from our methodology turned out to be clean, well structured, and easy to read.

Another interesting aspect of our approach is that the only parameter to be set is the number of refinement levels in the adaptive grid. In our tests we use three levels of refinement and we notice that the layout is not so easy to read when more than five levels are used. However,

Figure 40 – Visualizing 300 videos extracted from Youtube querying 6 different topics.



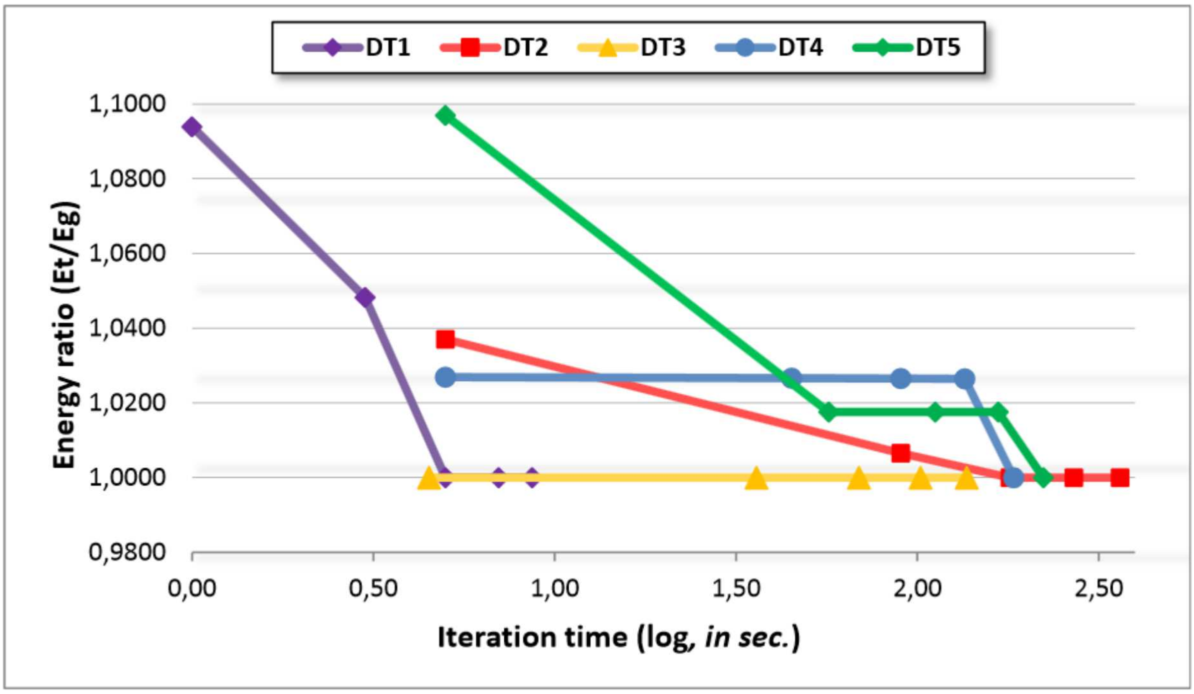
Source: Gomez-Nieto et al. (2016).

the appropriate number of refinement levels depends on the underlying application and the size of screen. A weakness of our methodology is that dense and tight data cluster may be cut into multiple cells during the grid construction. This problem can be mitigated through density-aware grid generation, an improvement to be incorporated in our methodology.

The quality of the semantic relation among grid cells also depends on the effectiveness of the multidimensional projection scheme in preserving neighborhoods during the mapping process. Visually encoding uncertainties as to neighborhood relation can help users during data exploration. However, this is an aspect that has not been properly addressed even in the context of multidimensional projection methods.

From our experimental analysis we also notice that it is not necessary to run the optimization procedure until convergence. In other words, good layouts are obtained after a few hundred iteration steps, being unnecessary to wait until a local minimum is reached. This fact makes the proposed methodology also attractive in terms of computational times. For instance, the layouts presented in Section 6.4 took less than six seconds to be produced. However, the algorithm can take a few minutes to fully converge if a local minimum is sought. To numerically illustrate this fact, we show in Fig. 41 the computational time spent by our methodology to reach a certain number of iterations versus the ratio between the resulting and global minimization energies E_t / E_g for each layout from Fig. 36. Note from the y-axis of the graph that how close to 1 the energy rates are, even for the layouts produced with a reduced number of iteration steps.

Figure 41 – Computational time versus optimization convergence rate for the layouts from Fig. 36.



Source: Gomez-Nieto et al. (2016).

6.6 Conclusion

We introduced a novel methodology to build geometric layouts to visualize data. In contrast to existing techniques, the proposed method makes use of a novel optimization procedure that is able to handle several requirements simultaneously, yielding structured overlap-free arrangements while still ensuring a semantic relation among neighbor entities. Furthermore, the method makes optimal use of display area, rendering it an effective and flexible tool for applications varying from image gallery construction to video data set visualization.

We are currently investigating interactive mechanisms to enable a free navigation throughout the layout as well as dynamic user-driven layout updates.

CHAPTER

7

SEMANTICALLY AWARE DYNAMIC LAYOUTS

7.1 Introduction

In this chapter 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.

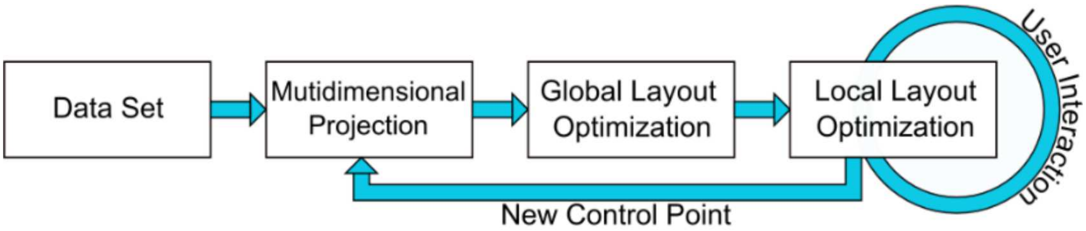
The provided results show the effectiveness of our approach in building and organizing user tailored layouts. Semantic relation between entities derives from similarity metrics, therefore, the proposed methodology can be employed in different scenarios and applications.

7.2 The method

The proposed mechanism to dynamically update layouts according to user intervention builds upon the methodology of ProjSnippet (GOMEZ-NIETO et al., 2014), which has been conceived to optimize layouts restricted to rectangular domains and with no interactive resources. Our approach, in contrast, enables interactive resources that allow users to 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 projections with an energy function tuned to enable interactive layout update as well as to enforce semantic relations among neighbor entities.

Figure 42 illustrates the main components of the proposed pipeline. From left to right,

Figure 42 – Method pipeline


Data Set
Mutidimensional Projection
Global Layout Optimization
Local Layout Optimization
User Interaction
New Control Point

Source: Gomez-Nieto, Motta e Nonato (2014).

given a data set and a similarity metric, the layout construction starts by mapping data instances to a visual space using a control point based multidimensional projection scheme, which forces similar instances to be projected close to each other. Each instance in the visual space is then associated to a geometric primitive (a box in our case), which are arranged in the visual space so as to avoid overlaps while preserving the neighborhood relations settled during the projection step. User can then modify the layout interactively by dragging and dropping geometric primitives. During user interaction a simplified optimization scheme is triggered to dynamically remove overlaps between instances, ensuring therefore a pleasant visual effect. When an instance selected by the user is released, it becomes a new control point for the multidimensional projection and the whole process is restarted from scratch. Since the initial control points are kept fixed, neighborhoods far from the region of interaction are mostly preserved by the multidimensional projection, maintaining the context on those parts of the layout.

The following subsections detail each step of the pipeline.

7.2.1 Building the initial layout

In our implementation we use LSP (PAULOVICH et al., 2008) as multidimensional projection mechanism. In order to keep the text self contained and better explain our approach we quickly describe the LSP method.

Let $P = \{p_1, \dots, p_n\}$ be a set of instances in a high-dimensional space and $G = \{P, E\}$ be the k-nearest neighbor graph of P, that is, each p_i (for simplicity we use the same notation p_i to represent instances and graph nodes) is a node of G and an edge $e_{ij} \in E$ if either p_j is a k-nearest neighbor of p_i or p_i is a k-nearest neighbor of p_j. Let L be the graph Laplacian matrix of G defined as $L = D - A$, where D is the diagonal matrix with $D_{ii} = \deg(p_i)$ (degree or valence of the node) and A is the adjacency matrix of G whose entry $A_{ij} = 1$ if there is an edge connecting p_i and p_j and $A_{ij} = 0$ otherwise.

Assuming G connected (we can ensure connectedness by adding new edges if necessary), the LSP method maps each instance p_i to the visual space by solving the two linear systems $L\mathbf{x} = 0$, $L\mathbf{y} = 0$ and setting the coordinates of $p_i = (x_i, y_i)$ where x_i and y_i are the i-th entry in the solution vectors $\tilde{x}$ and $\tilde{y}$.

The homogeneous systems $L\tilde{x} = 0$ and $L\tilde{y} = 0$ admit the trivial solutions $\tilde{x} = \tilde{y} = 0$. To avoid these trivial solutions constraints must be imposed to the systems, which is performed with the help of control points. Specifically, LSP assumes that a subset $P_c \subset P$, the control points, are endowed with coordinates in the visual space. The coordinates of the control points are used as constraints to the linear systems, thus avoiding the trivial solutions. Moreover, the control points constraints drive the whole mapping process. In other words, the whole projection follows the position of the control points, making instances that are neighbors of control points in the high-dimensional space to be mapped close to the corresponding control points in the visual space. The coordinates of the control points in the visual space are typically defined by a force scheme or manually specified by users.

The original LSP method imposes constraints using a least square based approach. Since the proposed interactive mechanism demands control points to be updated after each user interaction, we opt to impose constraints via the penalty method. More specifically, let $P_c = \{p_{i_1}, \ldots, p_{i_m}\}$ be the set of instances corresponding to control points, to use the penalty method the linear systems are then modified as follows:

$$(L + S)\tilde{x} = S\tilde{b}_x \quad \text{and} \quad (L + S)\tilde{y} = S\tilde{b}_y \tag{7.1}$$

where S is the penalty matrix with all entries equal zero except for the diagonal elements $S_{i_j i_j}$, $j = 1, \ldots, m$ corresponding to control point indexes, which receive to a large positive value, typically $S_{i_j i_j} = 10^8$. Vectors $\tilde{b}_x$ and $\tilde{b}_y$ also have entries equal zero except for the ones corresponding to the indexes of control points, which are set with the abscissa and ordinate coordinates of the control points, respectively. The advantage of using the penalty method is that only diagonal elements of L are affected when constraints are imposed. Therefore, L does not need to be recomputed every time a new constraint shows up.

Figure 43(a) illustrates the point cloud resulting from the projection step and Figure 43(b) shows the initial layout after associating geometric primitives (rectangular boxes) to each projected instance.

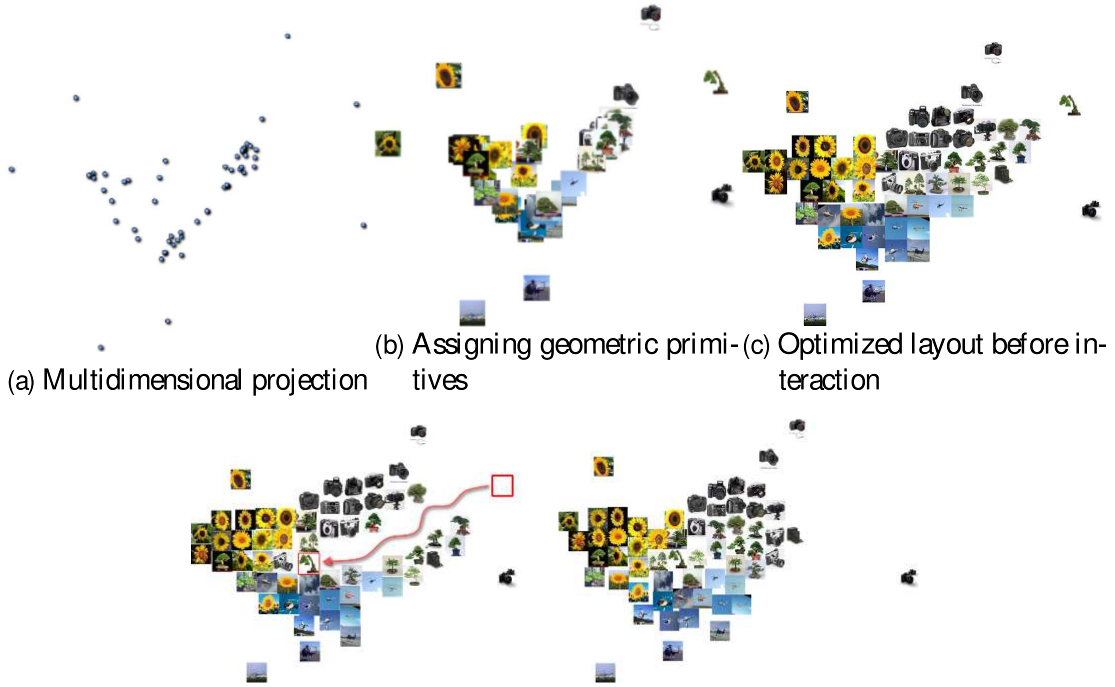
7.2.2 Semantic preserving overlap free layouts

At the end of the projection step geometric primitives associated to projected instances overlap considerably, impairing identification of individual entities and layout readability. In order to produce uncluttered layouts we optimize the position of geometric primitives to remove overlaps and preserve neighborhood structures established by the LSP.

The energy function that steers the layout arrangement is made up of two components

$$E = (1 - \alpha)E_O + \alpha E_N \tag{7.2}$$

Figure 43 – Visual result of each step of the proposed dynamic layout



(a) Multidimensional projection

(b) Assigning geometric primitives

(c) Optimized layout before interaction

(d) User interaction and local layout updates

(e) Optimized layout after interaction

Source: Gomez-Nieto, Motta e Nonato (2014).

The component E_O accounts for overlapping between primitives while E_N is related to neighborhood relations resulting from the projection step. The parameter $\alpha \in [0,1]$ balances the relative contribution of each component for the total energy E.

Energy E, as well as E_O and E_N, are functions of the coordinates of the bottom-left corners of the rectangles representing data entities. We omit the independent variables from the equations to simplify the notation.

Let $\vec{x}, \vec{y} \in R^n$ be the coordinate vectors of the bottom left corner of each rectangle and $\vec{v}, \vec{h} \in R^n$ be vectors whose components are the vertical and horizontal dimensions of each rectangle. The energy term E_O is then defined as:

$$E_O = \frac{2}{n(n+1)} \sum_{i=1}^{n} \sum_{j=i+1}^{n} \left[O_{i,j}(\vec{x},\vec{h})\, O_{i,j}(\vec{y},\vec{v}) \right] \tag{7.3}$$

where

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

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

where n is the number of data instances and 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 the function $O_{i,j}(\tilde{x},\tilde{h})$ is zero (minimal) if there is no horizontal overlapping between the rectangles i and j. The function $O_{i,j}(\tilde{y},\tilde{v})$ is defined similarly.

While E_O moves geometric primitives to prevent overlapping, the energy term E_N establishes semantic relations by forcing neighbor elements to be close to each other and it is defined from the Laplacian matrix L introduced in the last subsection. Denoting by $\tilde{x}^0$ and $\tilde{y}^0$ the x and y coordinate vectors of projected data instances we define the differential vectors δ_x and δ_y as:

$$\delta_x = L\tilde{x}^0, \qquad \delta_y = L\tilde{y}^0. \tag{7.4}$$

The neighborhood energy is then defined by:

$$E_N = \frac{n^2}{2\left(\|\delta_x\|^2 + \|\delta_y\|^2\right)}\left(\|L\tilde{x} - w\delta_x\|^2 + \|L\tilde{y} - w\delta_y\|^2\right). \tag{7.5}$$

E_N is minimal when the differentials $L\tilde{x}$ and $L\tilde{y}$ are proportional to the initial differentials δ_x and δ_y. In other words, the energy E_N is minimized when neighborhood relations are preserved during optimization. The unknown w is necessary to ensure that a scaled version of the originally projected layout is also a minimum of the neighborhood energy (w is optimized together with δ_x and δ_y). The factor $\frac{n^2}{2}\left(\|\delta_x\|^2 + \|\delta_y\|^2\right)^{-1}$ ensures that E_N has the same order of magnitude as E_O, thus both terms play similar roles in the total energy E. Figure 43(c) shows the optimized layout obtained from 43(b).

7.2.3 Enabling user interaction

There are two main issues that must be addressed to enable friendly interactive resources, namely, dynamic update of the layout during user interaction and preservation of semantic relations in the modified layout. Dynamically updating the layout during interaction is important to improve user experience and to ensure pleasant interaction resources. Preserving semantic relations is mandatory to facilitate layout readability and interpretation.

In order to dynamically update the layout during user interaction we trigger a simplified optimization scheme that removes overlaps between the primitive selected by the user and other primitives in the layout. Specifically, when the user drags a primitive, the layout is dynamically optimized, but using only the overlap energy E_O. The rationale behind temporarily neglecting the neighborhood energy term is the following: as most parts of the layout are not affected during interaction, the energy functional is already minimum in those regions, therefore, taking into account neighborhood energy E_N during interaction adds a unnecessary computational effort. Moreover, during interaction the user keeps attention on the object s/he is interacting with, neglecting semantic relations temporarily. Therefore, considering only E_O during interaction saves computational effort and this choice does not undermine user experience considerably.

The minimization of E_O can be done quite efficiently, as most parts of the layout are already in their minimum configuration during interaction. In fact, this minimization can be performed in interactive rates, enabling pleasant animation effects (see video in the supplemental material).

After interaction, we recover the semantic relation between neighbor instances by exploiting the flexibility provided by control points used in the multidimensional projection step. The idea is to make each user selected instance a new control point for the multidimensional projection. Once a selected instance is released by the user, its coordinates in the visual space are incorporated into vectors $\tilde{b}_x$ and $\tilde{b}_y$ in Equation (7.1), updating the penalty matrix S accordingly. The multidimensional projection is then run with the new set of control points and the layout rebuilt from scratch, that is, the minimization process is triggered according to equation (7.2). Since the original control points are fixed, most parts of the layout remain unchanged, preserving the visualization context as much as possible. Figures 43(d) and 43(e) illustrate how a layout is affected during and after user interaction.

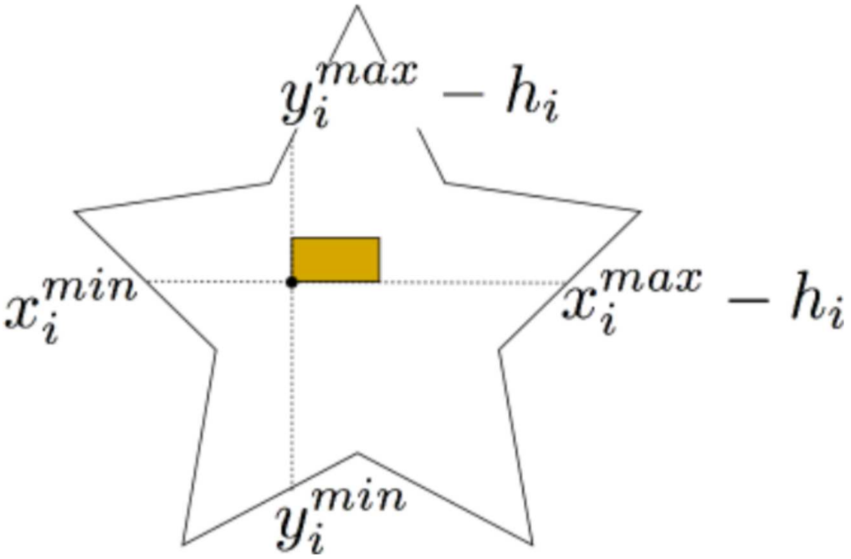
Replacing the layout modified during interaction by the layout updated after interaction would result in an abrupt visual change, making difficult to figure out which primitives has been affected due to user intervention. In order to provide a smoother transition between layouts we combine an interpolation scheme with overlap removal. Specifically, let L_b and L_a be the layouts resulting from user interaction and optimized with the new control points. Let l_{bi} and l_{ai} be the location of a given geometric primitive in L_b and L_a respectively. The smooth transition between l_{bi} and l_{ai} is performed as describe in the following algorithm:

Algorithm 1: Smooth transition

```
1  for $i = 1, \ldots, n$ do
2      $l_i \leftarrow l_{bi}$
3  repeat
4      for each $l_i$ do
5          $v_i \leftarrow l_{ai} - l_i$                          // displacement direction
6          $\delta_i \leftarrow \delta \|v_i\|$                    // displacement factor
7          $l_i \leftarrow l_i + \delta_i v_i / \|v_i\|$           // displacement towards $l_{ai}$
8      $\{l_i\} \leftarrow$ Overlap_Removal($\{l_i\}$)
9      $l_i \leftarrow (\|v_i\| - \delta_i) l_i / \|l_i\|$        // ensuring $l_i$ goes to $l_{ai}$
10 until $\max \|v_i\| \geq \varepsilon$;
```

Lines 1 to 2 initialize the smoothing transition process. Lines from 4 to 7 pushes all geometric primitives towards their final position in L_a. The overlap removal step in line 8 is necessary to ensure pleasant transition but it can push geometric primitives away from their final position. Line 9 ensures the primitives are indeed converging to their position in the final layout L_a. The scalar δ is fixed as 0.05, what makes δ_i to be five percent of the distance between l_i and l_{ai}. ε is fixed in 10^{-3}.

Figure 44 – Defining constraints to trap primitives within a giving visual domain



Source: Gomez-Nieto, Motta e Nonato (2014).

7.2.4 Dealing with arbitrary domains

In order to confine each geometric primitives within a given visual domain we impose constraints to the minimization problem. Specifically, the coordinates of the bottom left corner of each primitive are bounded by the visual domain boundary shape, as depicted in Figure 44 and mathematically formalized in equation (7.6).

$$\begin{aligned} \min \quad & (1-\alpha)E_O + \alpha E_N \\ \text{such that:} \quad & x_i^{min} \leq x_i \leq x_i^{max} - h_i, \quad i = 1,\ldots,n \\ & y_i^{min} \leq y_i \leq y_i^{max} - v_i, \quad i = 1,\ldots,n. \end{aligned} \tag{7.6}$$

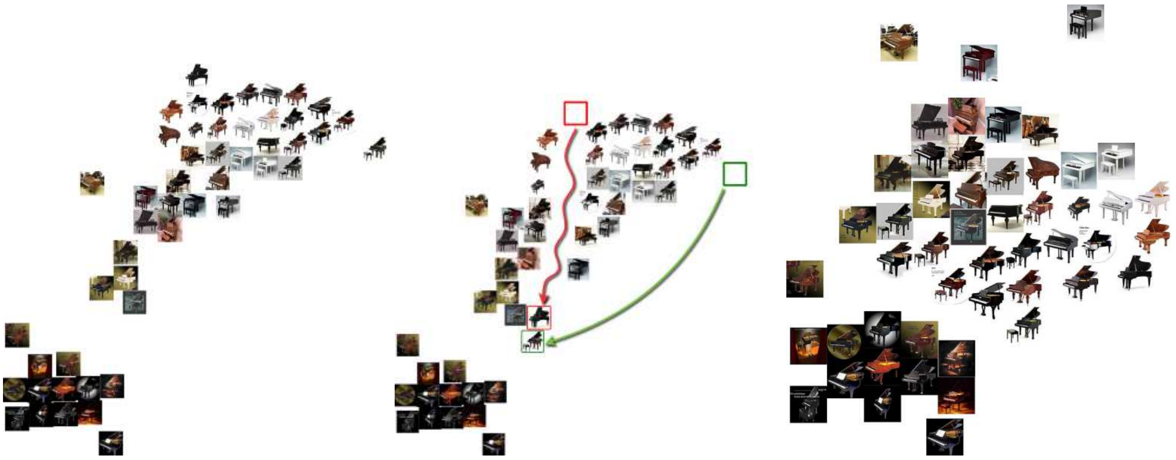
To speed up the optimization process we update the constraints only after a fixed number of optimization steps (20 in our implementation). Not updating the constraints after every optimization step allows primitives to move temporarily outside the visual domain. For visualization purposes, violating slightly the visual domain boundaries is not a problem and it is justified by the gain in terms of computational performance.

The minimization is accomplished by a globally convergent local optimization method, namely the Method of Moving Asymptotes (SVANBERG, 2002), available from the NLopt library at <http://ab-initio.mit.edu/wiki/index.php/NLopt>.

7.3 Results

In this section we present some layouts generated from the proposed methodology. All the results were produced using an Intel(R) Core(TM) i7-3537U computer with 8GB RAM.

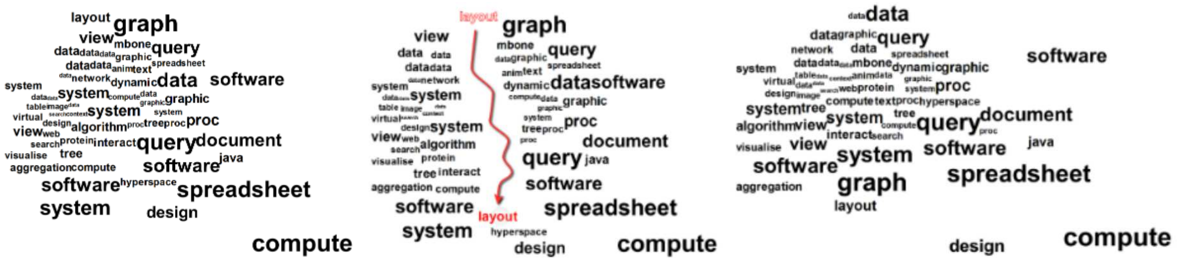
Figure 45 – Layouts produced from piano's images. From left to right: initial layout, layout during interaction, final layout



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

Source: Gomez-Nieto, Motta e Nonato (2014).

Figure 46 – Word cloud with size varying primitives

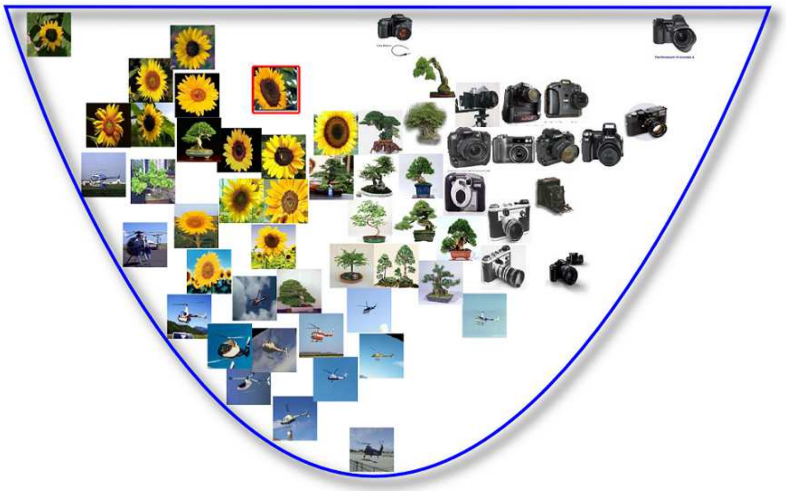


(a) Initial Layout (b) User Interaction (c) Final Layout

Source: Gomez-Nieto, Motta e Nonato (2014).

Figure 45 shows layouts generated from fifty images with dimensions 256 × 256 obtained from the Calltech 101 data set (FEI-FEI; FERGUS; PERONA, 2007). Those images where embedded in a feature space by extracting 96 features related to color moments. More specifically, each image was split in 16 non-overlapping regions from which the first and second statistical moments for each R,G, and B channel were extracted. Figure 45(a) shows the initial layout obtained by projecting and optimizing the layout according to equation (7.2) (a square box has been assigned to each projected image). Figure 45(b) show the layout during user interaction phase. The overlap removal opens a "path" throughout the region the user has dragged images. The opened path helps the user to keep a mental model of how much and where the layout is changing. Figure 45(c) depicts the final layout after optimization. Notice that the interaction changed the position of two white background piano images and, as a consequence, all pianos with white background has also been properly displaced, showing the capability of our approach in preserving semantic relations.

Figure 47 – Arbitrary Domain



Source: Gomez-Nieto, Motta e Nonato (2014).

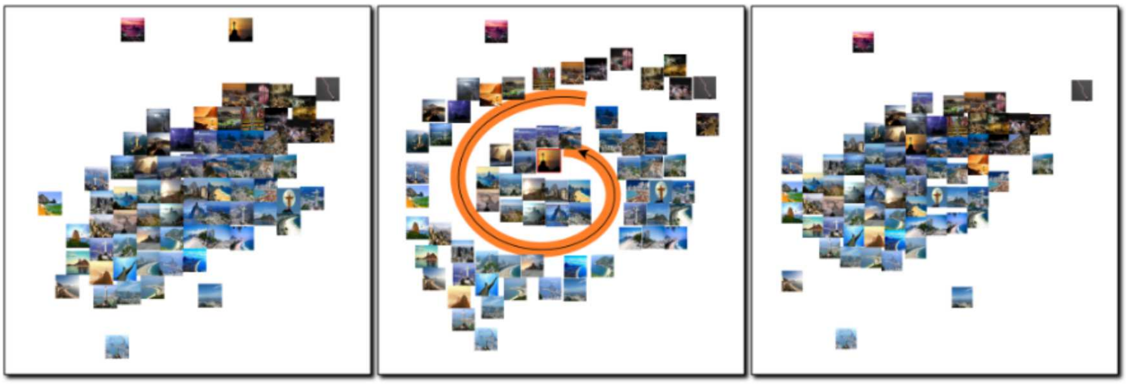
Figure 46 illustrates the use of our layout construction methodology in textual data analysis. More specifically, the layouts depicted in Figure 46 were generated by extracting bag-of-words from parts (abstract and title) of 60 visualization related papers published in the IEEE VisWeek Conference 2004. 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 46(a). Layouts generated during and after user intervention are depicted in Figures 46(b) and 46(c).

There are two important aspects to point out in this word cloud construction application. First, it shows the flexibility of our approach in handling geometric primitives with distinct sizes (in this case the rectangles bounding each word). Second, in contrast to other interactive schemes devoted to generate word clouds, our approach ensures a semantic relation between neighbor words in the cloud, which is highly desirable. The semantic relation here is defined by the similarity between documents measured from the word-frequency feature vector.

To illustrate how our approach behaves when facing irregular visual domains we present the layout depicted in Figure 47. Notice that despite the irregular shape of the underlying visual domain, our approach was able to generate a satisfactory result.

To conclude, we present in Figure 48 a layout generated from 70 images of Rio de Janeiro city obtained from searches in the web. The layout was drastically modified by a user (Figure 48

Figure 48 – Layout generated from 70 images of Rio de Janeiro city arranged with the proposed interactive layout construction mechanism.



Source: Gomez-Nieto, Motta e Nonato (2014).

middle) and our method was able to rebuild the layout respecting the similarity of the images (Figure 48 right). Notice that images of beaches are mostly concentrated at the bottom-right part of the layout while images related to the Christ's statue and Rio night life are on the center and top right parts of the layout.

7.4 Discussion, limitation and future Work

As shown in last section the proposed methodology enables free user interaction and it is able to produce pleasant layouts. Moreover, it turns out to be flexible enough to be employed in applications ranging from image gallery construction to word cloud based text visualization. In fact, the proposed methodology bears traits not present in most existing techniques such as the capability of preserving semantic relations in the layout.

When compared against state-of-art techniques as the one proposed by Reinert et al. (REINERT; RITSCHEL; SEIDEL, 2013), which also preserves semantic relations, our approach relies on a much simpler computational formulation/implementation. However, Reinert's technique has the advantage of producing pretty regular layouts, a property not present in our approach. Generating well structured layouts is a desirable property that facilitates readability and object location in the layout. How to modify the energy functional to produce more regular arrangements is a problem we intend to tackle in a future work.

Another drawback of our approach is related to local minima, mainly when dealing with very irregular visual domains. Local minima can lead to unsuitable layout configurations, hampering visualization. Although users can "reshape" the layout interactively, we are investigating alternatives that are less susceptible to local minima.

7.5 Conclusion

In this chapter we described a novel methodology to build layouts made up geometric primitives. Our approach is very flexible, allows free user intervention, and can deal with arbitrary visual domains, thus bearing traits not present in most existing methods. The proposed technique is useful in different visualization scenarios and quite effective in terms of semantic preservation during layout construction. The simple computational and mathematical formulation render our methodology a good alternative for tailoring semantic layouts.

CHAPTER

8

CONCLUSION AND FUTURE WORK

In this dissertation we presented four different formulations to address the generation of semantic layouts from multidimensional data. As can be noticed, all of these approaches deal simultaneously with multiple requirements for empowering exploration and analysis capabilities of similarity based 2D representations. Specifically, ProjSnippet (GOMEZ-NIETO et al., 2014) displays the results of textual queries applied to web search engines by using a novel overlap removal energy functional that considers both the neighborhood relations between instances and their overlapping in the visual space. Minimizing this energy functional provides a neighborhood preserving two-dimensional arrangement of the textual snippets with minimum overlap. MIOLA (GOMEZ-NIETO et al., 2013) uses a Mixed Integer Quadratic Formulation for overlap removal, enabling well structured layouts generation. It does not rely on intersection tests, making the algorithm simpler to implement. Moreover, it is quite flexible, being able to generate different layouts by just handling optimization constraints. Subsequently, we propose a new optimization method (GOMEZ-NIETO et al., 2016) that incorporates a wider range of requirements, i.e. display area optimization, hierarchical organization and object scaling. It allows to exploit efficiently the display space while preserve the scale ratio among entities, in addition to the set of requirements already addressed. Moreover, scalability is handled through a hierarchical representation scheme combined with navigation tools. Finally, our mechanism of semantically aware for dynamic updating (GOMEZ-NIETO; MOTTA; NONATO, 2014) allows free user intervention to rearrange the layout according to his/her preferences besides dealing efficiently with arbitrary visual domains.

One of the positive aspects in managing multiple requirements during layout construction is that we can play with properties such as object size and neighborhood to highlight relevant portions of the layout without losing the semantic relation among objects. By handling additional requisites such as overlap-free and optimal usage of display area we can reduce visual clutter while improving readability. As far as we known, no other techniques devoted to build layouts from geometric primitives is able to deal with so many concurrent requirements to generate meaningful layouts.

The practical usefulness of our methods is demonstrated in different multimedia data visualization applications. Within these we can mention textual snippets from web search engines, worclouds from document collections, image galleries and videos from video streaming sites. An important benefit of our methods is that are highly applicable to any dataset as long as it can be described as multidimensional data and a distance measure can be used to establish similarity relations between instances. A good example is the work proposed by Dasgupta et al. (2015) that improves readability of its visualization for climate model comparison using MIOLA.

Identifying new requirements is a mandatory task to evolve in this research topic. All of requirements addressed in this study were result of experimentation on multidimensional data associated to geometric primitives. As commented in Chapter 1, different types of data requires of different designs and functionalities for extracting as much information as possible, in this way we believe that introducing semantic layouts into different contexts that to the focused in this work will widely enrich the development of this topic.

We are currently investigating new interactive mechanisms to enable a free navigation throughout the layout as well as dynamic user-driven layout updates. A major drawback is concerns to the number of instances to optimize during semantic interactions since solving an optimization problem with a high number of variables fast enough to provide a real time interaction is computationally costly.

BIBLIOGRAPHY

AMORIM, E. P. dos S.; BRAZIL, E. V.; DANIELS, J.; JOIA, P.; NONATO, L. G.; SOUSA, M. C. ilamp: Exploring high-dimensional spacing through backward multidimensional projection. In: Visual Analytics Science and Technology (VAST), 2012 IEEE Conference on. [S.l.: s.n.], 2012. p. 53–62. Cited on page 34.

ARTHUR, D.; VASSILVITSKII, S. k-means++: the advantages of careful seeding. In: ACM-SIAM SODA. [S.l.: s.n.], 2007. p. 1027–1035. Cited on page 57.

BALZER, M.; DEUSSEN, O.; LEWERENTZ, C. Voronoi treemaps for the visualization of software metrics. In: Proceedings of the ACM Symposium on Software Visualization. New York, NY, USA: ACM, 2005. (SoftVis'2005), p. 165–172. ISBN 1-59593-073-6. Cited 2 times on pages 50 and 53.

BEDERSON, B. B.; SHNEIDERMAN, B.; WATTENBERG, M. Ordered and quantum treemaps: Making effective use of 2d space to display hierarchies. ACM Transactions on Graphics (TOG), ACM, v. 21, n. 4, p. 833–854, 2002. Cited 2 times on pages 50 and 53.

BERTIN, J. Graphics and graphic information-processing. [S.l.]: de Gruyter (The University of Michigan), 2007. ISBN 9783110088687. Cited 3 times on pages 27, 50, and 51.

BLANCH, R.; LECOLINET, E. Browsing zoomable treemaps: Structure-aware multi-scale navigation techniques. IEEE Transactions on Visualization and Computer Graphics, IEEE Computer Society, v. 13, n. 6, p. 1248–1253, 2007. Cited on page 52.

BRULS, M.; HUIZING, K.; WIJK, J. J. V. Squarified treemaps. In: Data Visualization. [S.l.]: Springer, 2000. p. 33–42. Cited on page 53.

BUCHIN, K.; EPPSTEIN, D.; LÖFFLER, M.; NÖLLENBURG, M.; SILVEIRA, R. I. Adjacency-preserving spatial treemaps. In: ____. Algorithms and Data Structures: 12th International Symposium, WADS 2011, New York, NY, USA, August 15-17, 2011. Proceedings. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011. p. 159–170. ISBN 978-3-642-22300-6. Cited 2 times on pages 50 and 52.

CHUANG, J.-H.; LIN, C.-C.; YEN, H.-C. Drawing graphs with nonuniform nodes using potential fields. In: SPRINGER. Graph Drawing. [S.l.], 2004. Cited 2 times on pages 49 and 50.

CUTTING, D. R.; KARGER, D. R.; PEDERSEN, J. O.; TUKEY, J. W. Scatter/gather: A cluster-based approach to browsing large document collections. In: ACM SIGIR. [S.l.: s.n.], 1992. p. 318–329. Cited on page 57.

DANIELS, J.; ANDERSON, E. W.; NONATO, L. G.; SILVA, C. T. Interactive vector field feature identification. IEEE Transactions on Visualization and Computer Graphics, v. 16, n. 6, p. 1560–1568, Nov 2010. ISSN 1077-2626. Cited on page 34.

DASGUPTA, A.; POCO, J.; WEI, Y.; COOK, R.; BERTINI, E.; SILVA, C. T. Bridging theory with practice: An exploratory study of visualization use and design for climate model comparison.

IEEE Transactions on Visualization and Computer Graphics, v. 21, n. 9, p. 996–1014, Sept 2015. ISSN 1077-2626. Cited on page 122.

DO, M. N.; VETTERLI, M. Framing pyramids. IEEE Transactions on Signal Processing, v. 51, n. 9, p. 2329–2342, 2003. Cited on page 102.

DU, Q.; FABER, V.; GUNZBURGER, M. Centroidal voronoi tessellations: Applications and algorithms. SIAM Review, v. 41, n. 4, p. 637, 1999. Cited 2 times on pages 66 and 84.

DUARTE, F. S. L. G.; SIKANSI, F.; FATORE, F. M.; FADEL, S. G.; PAULOVICH, F. V. Nmap: A novel neighborhood preservation space-filling algorithm. IEEE Transactions on Visualization and Computer Graphics, v. 99, 2014. ISSN 1077-2626. Cited 2 times on pages 50 and 52.

DWYER, T.; MARRIOTT, K.; STUCKEY, P. J. Fast node overlap removal. In: SPRINGER. Graph Drawing. [S.l.], 2006. p. 153–164. Cited 5 times on pages 49, 50, 66, 84, and 101.

DWYER, T.; MEARS, C.; MORGAN, K.; NIVEN, T.; MARRIOTT, K.; WALLACE, M. Improved optimal and approximate power graph compression for clearer visualisation of dense graphs. In: 2014 IEEE Pacific Visualization Symposium. [S.l.: s.n.], 2014. p. 105–112. ISSN 2165-8765. Cited on page 36.

DWYER, T.; NACHMANSON, L. Fast edge-routing for large graphs. In: SPRINGER. Graph Drawing. [S.l.], 2010. p. 147–158. Cited 2 times on pages 50 and 51.

DWYER, T.; RICHE, N. H.; MARRIOTT, K.; MEARS, C. Edge compression techniques for visualization of dense directed graphs. IEEE Transactions on Visualization and Computer Graphics, IEEE Computer Society, Los Alamitos, CA, USA, v. 19, n. 12, p. 2596–2605, 2013. ISSN 1077-2626. Cited on page 36.

ELZEN, S. Van den; WIJK, J. J. van. Small multiples, large singles: A new approach for visual data exploration. Computer Graphics Forum, Wiley-Blackwell Publishing Ltd, v. 32, n. 3pt2, p. 191–200, 2013. ISSN 1467-8659. Cited 3 times on pages 27, 50, and 51.

ESTER, M.; KRIEGEL, H.-P.; SANDER, J.; XU, X. A density-based algorithm for discovering clusters in large spatial databases with noise. In: . [S.l.]: AAAI Press, 1996. p. 226–231. Cited 2 times on pages 38 and 42.

FALOUTSOS, C.; LIN, K.-I. Fastmap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. SIGMOD Rec., ACM, New York, NY, USA, v. 24, n. 2, p. 163–174, maio 1995. ISSN 0163-5808. Cited on page 34.

FEI-FEI, L.; FERGUS, R.; PERONA, P. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. Computer Vision and Image Understanding, Elsevier, v. 106, n. 1, p. 59–70, 2007. Cited 2 times on pages 102 and 116.

FEKETE, J.-D.; PLAISANT, C. Interactive information visualization of a million items. In: IEEE COMPUTER SOCIETY. IEEE Symposium on Information Visualization (INFOVIS 2002). [S.l.], 2002. p. 117–124. Cited 2 times on pages 27 and 53.

FRIED, O.; DIVERDI, S.; HALBER, M.; SIZIKOVA, E.; FINKELSTEIN, A. IsoMatch: Creating informative grid layouts. Computer Graphics Forum (Proc. Eurographics), v. 34, n. 2, maio 2015. Cited 4 times on pages 50, 51, 101, and 103.

GANSNER, E. R.; HU, Y. Graph drawing. In: TOLLIS, I. G.; PATRIGNANI, M. (Ed.). Berlin, Heidelberg: Springer-Verlag, 2009. cap. Efficient Node Overlap Removal Using a Proximity Stress Model, p. 206–217. ISBN 978-3-642-00218-2. Cited 5 times on pages 50, 51, 66, 84, and 101.

GASSEN, S. V.; CALLEBAUT, B.; HELDEN, M. J. V.; LAMBRECHT, B. N.; DEMEESTER, P.; DHAENE, T.; SAEYS, Y. Flowsom: Using self-organizing maps for visualization and interpretation of cytometry data. Cytometry Part A, v. 87, n. 7, p. 636–645, 2015. ISSN 1552-4930. Cited on page 35.

GOMEZ-NIETO, E.; CASACA, W.; MOTTA, D.; HARTMANN, I.; TAUBIN, G.; NONATO, L. G. Dealing with multiple requirements in geometric arrangements. IEEE Transactions on Visualization and Computer Graphics, v. 22, n. 3, p. 1223–1235, March 2016. ISSN 1077-2626. Cited 13 times on pages 29, 92, 93, 95, 96, 97, 99, 100, 103, 104, 106, 107, and 121.

GOMEZ-NIETO, E.; CASACA, W.; NONATO, L. G.; TAUBIN, G. Mixed integer optimization for layout arrangement. In: Proceedings of the 26th Conference on Graphics, Patterns and Images. Washington, DC, USA: IEEE Computer Society, 2013. (SIBGRAPI'13), p. 115–122. ISBN 978-0-7695-5099-2. Cited 13 times on pages 29, 78, 81, 82, 83, 85, 86, 87, 88, 89, 98, 101, and 121.

GOMEZ-NIETO, E.; MOTTA, D.; NONATO, L. Semantically aware dynamic layouts. In: Graphics, Patterns and Images (SIBGRAPI), 2014 27th SIBGRAPI Conference on. [S.l.: s.n.], 2014. p. 220–226. Cited 8 times on pages 29, 110, 112, 115, 116, 117, 118, and 121.

GOMEZ-NIETO, E.; ROMAN, F. S.; PAGLIOSA, P.; CASACA, W.; HELOU, E. S.; OLIVEIRA, M. C. F. de; NONATO, L. G. Similarity preserving snippet-based visualization of web search results. IEEE Transactions on Visualization and Computer Graphics, IEEE Computer Society, Los Alamitos, CA, USA, v. 20, n. 3, p. 457–470, 2014. ISSN 1077-2626. Cited 17 times on pages 29, 57, 59, 61, 62, 64, 65, 66, 67, 68, 70, 71, 72, 73, 101, 109, and 121.

HAREL, D.; KOREN, Y. Drawing graphs with non-uniform vertices. In: ACM. Proceedings of the Working Conference on Advanced Visual Interfaces. [S.l.], 2002. p. 157–166. Cited 2 times on pages 49 and 50.

HARROWER, M.; BREWER, C. A. ColorBrewer.org: An Online Tool for Selecting Colour Schemes for Maps. The Cartographic Journal, Maney Publishing, v. 40, n. 1, p. 27–37, jun. 2003. ISSN 0008-7041. Cited on page 38.

HEALEY, C.; ENNS, J. Attention and visual memory in visualization and computer graphics. IEEE Transactions on Visualization and Computer Graphics, v. 18, n. 7, p. 1170–1188, July 2012. ISSN 1077-2626. Cited on page 37.

HOFFMAN, P.; GRINSTEIN, G.; MARX, K.; GROSSE, I.; STANLEY, E. Dna visual and analytic data mining. In: Visualization '97., Proceedings. [S.l.: s.n.], 1997. p. 437–441. Cited on page 37.

HUANG, X.; LAI, W.; SAJEEV, A.; GAO, J. A new algorithm for removing node overlapping in graph visualization. Information Sciences, Elsevier, v. 177, n. 14, p. 2821–2844, 2007. Cited on page 49.

INSELBERG, A. A survey of parallel coordinates. In: ____. Mathematical Visualization: Algorithms, Applications and Numerics. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998. p. 167–179. ISBN 978-3-662-03567-2. Cited 2 times on pages 37 and 40.

JANSEN, K.; PRADEL, L. How to Maximize the Total Area of Rectangles Packed into a Rectrangle? [S.l.]: Inst. fur Informatik, Cristian-Albrechts-Univ., Kiev, 2009. Cited on page 95.

JAVED, W.; MCDONNEL, B.; ELMQVIST, N. Graphical perception of multiple time series. IEEE Transactions on Visualization and Computer Graphics, v. 16, n. 6, p. 927–934, 2010. ISSN 1077-2626. Cited 2 times on pages 50 and 51.

JEONG, D. H.; ZIEMKIEWICZ, C.; FISHER, B.; RIBARSKY, W.; CHANG, R. ipca: An interactive system for pca-based visual analytics. In: Proceedings of the 11th Eurographics / IEEE - VGTC Conference on Visualization. Chichester, UK: The Eurographs Association & #38; John Wiley & #38; Sons, Ltd., 2009. (EuroVis'09), p. 767–774. Cited on page 34.

JIANG, H.; NAN, L.; YAN, D.; DONG, W.; ZHANG, X.; WONKA, P. Automatic constraint detection for 2d layout regularization. IEEE Transactions on Visualization and Computer Graphics, PP, n. 99, p. 1–1, 2015. ISSN 1077-2626. Cited 2 times on pages 50 and 52.

JOHNSON, B.; SHNEIDERMAN, B. Tree-maps: A space-filling approach to the visualization of hierarchical information structures. In: Proceedings of the 2Nd Conference on Visualization. Los Alamitos, CA, USA: IEEE Computer Society Press, 1991. (VIS'91), p. 284–291. ISBN 0-8186-2245-8. Cited on page 52.

JOIA, P.; COIMBRA, D.; CUMINATO, J.; PAULOVICH, F.; NONATO, L. Local affine multidimensional projection. IEEE Trans. on Visualization and Computer Graphics, v. 17, p. 2563–2571, 2011. Cited 3 times on pages 34, 67, and 86.

JOIA, P.; GOMEZ-NIETO, E.; NETO, J. B.; CASACA, W.; BOTELHO, G.; PAIVA, A.; NONATO, L. G. Class-specific metrics for multidimensional data projection applied to cbir. The Visual Computer, Springer Berlin / Heidelberg, v. 28, p. 1027–1037, 2012. ISSN 0178-2789. Cited on page 34.

JULESZ, B. Textons, the elements of texture perception, and their interactions. Nature, v. 290, n. 5802, p. 91–97, Mar 1981. Cited on page 38.

____. A brief outline of the texton theory of human vision. Trends in Neurosciences, v. 7, n. 2, p. 41 – 45, 1984. ISSN 0166-2236. Cited on page 38.

KANDOGAN, E. Visualizing multi-dimensional clusters, trends, and outliers using star coordinates. In: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, NY, USA: ACM, 2001. (KDD '01), p. 107–116. ISBN 1-58113-391-X. Cited 2 times on pages 37 and 40.

KEHRER, J.; PIRINGER, H.; BERGER, W.; GROLLER, M. E. A model for structure-based comparison of many categories in small-multiple displays. IEEE Transactions on Visualization and Computer Graphics, IEEE Computer Society, v. 19, n. 12, p. 2287–2296, 2013. Cited 2 times on pages 50 and 51.

KEIM, D.; ZHANG, L. Solving problems with visual analytics: Challenges and applications. In: Proceedings of the 11th International Conference on Knowledge Management and Knowledge Technologies. New York, NY, USA: ACM, 2011. (i-KNOW '11), p. 1:1–1:4. ISBN 978-1-4503-0732-1. Cited on page 39.

KIEFFER, S.; DWYER, T.; MARRIOTT, K.; WYBROW, M. Incremental grid-like layout using soft and hard constraints. In: Graph Drawing (Lectures Notes in Computer Science). [S.l.: s.n.], 2013. p. 448–459. Cited 2 times on pages 49 and 50.

KOH, K.; LEE, B.; KIM, B.; SEO, J. Maniwordle: Providing flexible control over wordle. IEEE Transactions on Visualization and Computer Graphics, IEEE Educational Activities Department, Piscataway, NJ, USA, v. 16, n. 6, p. 1190–1197, nov. 2010. ISSN 1077-2626. Cited 3 times on pages 27, 50, and 53.

KOHONEN, T. Self-organized formation of topologically correct feature maps. Biological Cybernetics, v. 43, n. 1, p. 59–69, 1982. ISSN 1432-0770. Cited on page 35.

KÖNIG, A. Interactive visualization and analysis of hierarchical neural projections for data mining. IEEE Transactions on Neural Networks, IEEE, v. 11, n. 3, p. 615–624, 2000. Cited on page 45.

LAGUS, K.; KASKI, S.; KOHONEN, T. Mining massive document collections by the websom method. Inf. Sci., Elsevier Science Inc., New York, NY, USA, v. 163, n. 1-3, p. 135–156, 2004. ISSN 0020-0255. Cited on page 35.

LAZIMY, R. Mixed-integer quadratic programming. Maht. Programming, v. 22, p. 332–349, 1982. Cited on page 79.

LEE, J.; LEYFFER, S. Mixed Integer Nonlinear Programming (The IMA Volumes in Mathematics and its Applications). [S.l.]: Springer, 2012. Cited 2 times on pages 83 and 98.

LEHMANN, D. J.; THEISEL, H. General projective maps for multidimensional data projection. Computer Graphics Forum, v. 35, n. 2, p. 443–453, 2016. ISSN 1467-8659. Cited on page 34.

LEUNG, J.; TAM, T.; WONG, C.; YOUNG, G.; CHIN, F. Packing squares into a square. Journal of Parallel and Distributed Computing, Elsevier, v. 10, n. 3, p. 271–275, 1990. Cited on page 94.

LIERE, R. V.; LEEUW, W. D. Graphsplatting: Visualizing graphs as continuous fields. IEEE Transactions on Visualization and Computer Graphics, IEEE, v. 9, n. 2, p. 206–212, 2003. Cited on page 40.

LIU, X.; HU, Y.; NORTH, S.; LEE, T.-Y.; SHEN, H.-W. CorrelatedMultiples: Spatially Coherent Small Multiples with Constrained Multidimensional Scaling. Tr10. [S.l.], 2013. 1–10 p. Cited 2 times on pages 50 and 51.

LLOYD, S. Least squares quantization in pcm. IEEE Transactions on Information Theory, v. 28, n. 2, p. 129–137, Mar 1982. ISSN 0018-9448. Cited 2 times on pages 38 and 42.

LUHN, H. P. The automatic creation of literature abstracts. IBM Journal of research and development, IBM, v. 2, n. 2, p. 159–165, 1958. Cited 2 times on pages 56 and 102.

MARRIOTT, K.; STUCKEY, P.; TAM, V.; HE, W. Removing node overlapping in graph layout using constrained optimization. Constraints, Springer, v. 8, n. 2, p. 143–171, 2003. Cited 2 times on pages 49 and 50.

MISUE, K.; EADES, P.; LAI, W.; SUGIYAMA, K. Layout adjustment and the mental map. Journal of visual languages and computing, Elsevier, v. 6, n. 2, p. 183–210, 1995. Cited 3 times on pages 45, 49, and 50.

MURPHY, R. D.; ABERE, M. J.; ZHANG, H.; SUN, H.; TORRALVA, B.; MANSFIELD, J. F.; KOTOV, N. A.; YALISOVE, S. M. Ultrafast laser orthogonal alignment and patterning of carbon nanotube-polymer composite films. Applied Physics Letters, v. 101, n. 20, p. 203 – 301, 2012. Cited on page 52.

MUSIALSKI, P.; WONKA, P.; ALIAGA, D. G.; WIMMER, M.; GOOL, L. van; PURGATHOFER, W. A survey of urban reconstruction. Computer Graphics Forum, v. 32, n. 6, p. 146–177, 2013. ISSN 1467-8659. Cited on page 52.

NISHIDA, G.; GARCIA-DORADO, I.; ALIAGA, D. G.; BENES, B.; BOUSSEAU, A. Interactive sketching of urban procedural models. ACM Trans. Graph., v. 35, n. 4, 2016. Cited on page 52.

NOCAJ, A.; BRANDES, U. Organizing search results with a reference map. IEEE Transactions on Visualization and Computer Graphics, v. 18, n. 12, p. 2546–2555, 2012. Cited 2 times on pages 50 and 53.

PAGLIOSA, P.; MARTINS, R. M.; CEDRIM, D.; PAIVA, A.; MINGHIM, R.; NONATO, L. G. Mist: Multiscale information and summaries of texts. In: Sibgrapi 2013 (26th Conference on Graphics, Patterns and Images). [S.l.: s.n.], 2013. p. 91–98. ISSN 1530-1834. Cited on page 34.

PAULOVICH, F. V.; MINGHIM, R. Hipp: A novel hierarchical point placement strategy and its application to the exploration of document collections. IEEE Transactions on Visualization and Computer Graphics, v. 14, n. 6, p. 1229–1236, Nov 2008. ISSN 1077-2626. Cited on page 45.

PAULOVICH, F. V.; NONATO, L. G.; MINGHIM, R.; LEVKOWITZ, H. Least square projection: A fast high-precision multidimensional projection technique and its application to document mapping. IEEE Transactions on Visualization and Computer Graphics, v. 14, n. 3, p. 564–575, 2008. Cited 7 times on pages 34, 57, 83, 91, 101, 102, and 110.

PAULOVICH, F. V.; SILVA, C. T.; NONATO, L. G. Two-phase mapping for projecting massive data sets. IEEE Transactions on Visualization and Computer Graphics, v. 16, n. 6, p. 1281–1290, 2010. Cited on page 34.

PAULOVICH, F. V.; TOLEDO, F.; TELLES, G. P.; MINGHIM, R.; NONATO, L. G. Semantic wordification of document collections. In: WILEY-BLACKWELL ONLINE LIBRARY. Computer Graphics Forum. [S.l.], 2012. v. 31, n. 3pt3, p. 1145–1153. Cited 3 times on pages 27, 103, and 105.

PINHO, R. D.; OLIVEIRA, M. C.; LOPES, A. A. An incremental space to visualize dynamic data sets. Multimedia Tools and Applications, Kluwer Academic Publishers, Hingham, MA, USA, v. 50, n. 3, p. 533–562, 2010. ISSN 1380-7501. Cited 6 times on pages 27, 43, 50, 51, 101, and 103.

PIROLLI, P.; SCHANK, P.; HEARST, M.; DIEHL, C. Scatter/gather browsing communicates the topic structure of a very large text collection. In: ACM SIGCHI. [S.l.: s.n.], 1996. p. 213–220. Cited on page 57.

REINERT, B.; RITSCHEL, T.; SEIDEL, H.-P. Interactive by-example design of artistic packing layouts. ACM Transactions on Graphics (TOG), ACM, v. 32, n. 6, p. 218, 2013. Cited 4 times on pages 43, 50, 53, and 118.

RUEGG, U.; KIEFFER, S.; DWYER, T.; MARRIOTT, K.; WYBROW, M. Stress-minimizing orthogonal layout of data flow diagrams with ports. In: Graph Drawing. [S.l.: s.n.], 2014. p. 1–16. Cited 2 times on pages 49 and 50.

SALTON, G. Developments in automatic text retrieval. Science, v. 253, p. 974–980, 1991. Cited 2 times on pages 56 and 102.

SALTON, G.; WONG, A.; YANG, C. S. A vector space model for automatic indexing. Commun. ACM, ACM, New York, NY, USA, v. 18, n. 11, p. 613–620, 1975. ISSN 0001-0782. Cited on page 83.

SAMET, H. Foundations of Multidimensional and Metric Data Structures. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2005. Cited on page 46.

SCHöLKOPF, B.; SMOLA, A.; MüLLER, K.-R. Nonlinear component analysis as a kernel eigenvalue problem. Neural Comput., MIT Press, Cambridge, MA, USA, v. 10, n. 5, p. 1299–1319, jul. 1998. ISSN 0899-7667. Cited on page 34.

SEYMOUR, D.; BRITTON, J. Introduction to Tessellations. [S.l.]: Dale Seymour Publications, 1989. ISBN 9780866514613. Cited on page 38.

SHNEIDERMAN, B. Tree visualization with tree-maps: 2-d space-filling approach. ACM Trans. Graph., ACM, New York, NY, USA, v. 11, n. 1, p. 92–99, jan. 1992. ISSN 0730-0301. Cited 2 times on pages 50 and 52.

____. Treemaps for space-constrained visualization of hierarchies. 1998–2013. Disponível em: <http://www.cs.umd.edu/hcil/treemap-history/>. Cited on page 53.

SHNEIDERMAN, B.; WATTENBERG, M. Ordered treemap layouts. In: IEEE COMPUTER SOCIETY. IEEE Symposium on Information Visualization (INFOVIS 2001). [S.l.], 2001. p. 73–73. Cited on page 53.

SPRITZER, A. S.; FREITAS, C. M. D. S. Design and evaluation of magnetviz—a graph visualization tool. Visualization and Computer Graphics, IEEE Transactions on, IEEE, v. 18, n. 5, p. 822–835, 2012. Cited on page 49.

STAHNKE, J.; DöRK, M.; MüLLER, B.; THOM, A. Probing projections: Interaction techniques for interpreting arrangements and errors of dimensionality reductions. IEEE Transactions on Visualization and Computer Graphics, v. 22, p. 629–638, 2016. Cited on page 34.

STROBELT, H.; SPICKER, M.; STOFFEL, A.; KEIM, D.; DEUSSEN, O. Rolled-out wordles: A heuristic method for overlap removal of 2d data representatives. Computer Graphics Forum, Wiley-Blackwell Publishing Ltd, v. 31, n. 3pt3, p. 1135–1144, 2012. ISSN 1467-8659. Cited 7 times on pages 27, 44, 50, 51, 66, 84, and 101.

STRONG, G.; GONG, M. Data organization and visualization using self-sorting map. In: Proceedings of Graphics Interface 2011. [S.l.]: Canadian Human-Computer Communications Society, 2011. (GI'2011), p. 199–206. ISBN 978-1-4503-0693-5. Cited 2 times on pages 27 and 51.

____. Self-sorting map: An efficient algorithm for presenting multimedia data in structured layouts. Multimedia, IEEE Transactions on, v. 16, n. 4, p. 1045–1058, June 2014. ISSN 1520-9210. Cited 6 times on pages 27, 35, 50, 51, 101, and 103.

SVANBERG, K. A class of globally convergent optimization methods based on conservative convex separable approximations. SIAM Journal on Optimization, v. 12, n. 2, p. 555–573, 2002. Cited 2 times on pages 60 and 115.

TEEVAN, J.; CUTRELL, E.; FISHER, D.; DRUCKER, S. M.; RAMOS, G.; ANDRé, P.; HU, C. Visual snippets: summarizing web pages for search and revisitation. In: ACM SIGCHI. [S.l.: s.n.], 2009. p. 2023–2032. Cited on page 55.

TEJADA, E.; MINGHIM, R.; NONATO, L. G. On improved projection techniques to support visual exploration of multidimensional data sets. Information Visualization, Palgrave Macmillan, v. 2, n. 4, p. 218–231, dez. 2003. ISSN 1473-8716. Cited on page 34.

TORGERSON, W. S. Multidimensional scaling: I. theory and method. Psychometrika, v. 17, p. 401–419, 1952. Cited on page 34.

TUCKER, A. The Computer Science and Engineering Handbook. [S.l.]: CRC-Press, 1996. ISBN 9780849329098. Cited on page 40.

TUFTE, E. Envisioning information. [S.l.]: Graphics Press, 1990. (Envisioning Information, 914). Cited 3 times on pages 27, 50, and 51.

TUTTLE, C.; NONATO, L. G.; SILVA, C. Pedvis: A structured, space-efficient technique for pedigree visualization. IEEE Transactions on Visualization and Computer Graphics, IEEE Computer Society, v. 16, n. 6, p. 1063–1072, 2010. ISSN 1077-2626. Cited 2 times on pages 50 and 53.

VALLET, J.; MELANÇON, G.; PINAUD, B. Jasper: Just a new space-filling and pixel-oriented layout for large graph overview. Electronic Imaging, v. 2016, n. 1, p. 1–10, 2016. Cited 2 times on pages 50 and 52.

WARE, C. Information Visualization: Perception for Design. [S.l.]: Elsevier Science, 2004. (Interactive Technologies). ISBN 9780080478494. Cited on page 37.

WOOD, J.; DYKES, J. Spatially ordered treemaps. IEEE Transactions on Visualization and Computer Graphics, IEEE Computer Society, v. 14, n. 6, p. 1348–1355, 2008. Cited 2 times on pages 50 and 53.

WU, Y.; PROVAN, T.; WEI, F.; LIU, S.; MA, K.-L. Semantic-preserving word clouds by seam carving. Computer Graphics Forum, v. 30, n. 3, p. 741–750, 2011. Cited on page 60.

XU, P.; FU, H.; IGARASHI, T.; TAI, C.-L. Global beautification of layouts with interactive ambiguity resolution. In: Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology. New York, NY, USA: ACM, 2014. (UIST '14), p. 243–252. ISBN 978-1-4503-3069-5. Cited 2 times on pages 50 and 52.

XU, P.; FU, H.; TAI, C.-L.; IGARASHI, T. Gaca: Group-aware command-based arrangement of graphic elements. In: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems. New York, NY, USA: ACM, 2015. (CHI '15), p. 2787–2795. ISBN 978-1-4503-3145-6. Cited 2 times on pages 50 and 52.

XU, R.; WUNSCH, D. Clustering (IEEE Press Series on Computational Intelligence). [S.l.]: IEEE Press, Wiley, NJ, 2009. Cited on page 46.

YIN, W. Gurobi mex: A matlab interface for gurobi (2011). In: convexoptimization.com/wikimization/index.php/gurobi_mex. [S.l.: s.n.], 2011. Cited on page 83.

YOGHOURDJIAN, V.; DWYER, T.; GANGE, G.; KIEFFER, S.; KLEIN, K.; MARRIOTT, K. High-quality ultra-compact grid layout of grouped networks. IEEE Transactions on Visualization and Computer Graphics, v. 22, n. 1, p. 339 – 348, 1 2016. ISSN 1077-2626. Cited 2 times on pages 36 and 40.