DOI: 10.1111/j.1467-8659.2012.03106.x
Eurographics/ IEEE-VGTC Conference on Visualization 2012
S. Bruckner, S. Miksch, and H. Pfister
(Guest Editors)

Volume 31 (2012), Number 3

Rolled-out Wordles: A Heuristic Method for Overlap Removal of 2D Data Representatives

H. Strobelt, M. Spicker, A. Stoffel, D. Keim, O. Deussen

University of Konstanz

Abstract

When representing 2D data points with spacious objects such as labels, overlap can occur. We present a simple algorithm which modifies the (Mani-)Wordle idea with scan-line based techniques to allow a better placement. We give an introduction to common placement techniques from different fields and compare our method to these techniques w.r.t. euclidean displacement, changes in orthogonal ordering as well as shape and size preservation. Especially in dense scenarios our method preserves the overall shape better than known techniques and allows a good trade-off between the other measures. Applications on real world data are given and discussed.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Line and curve generation

1. Introduction

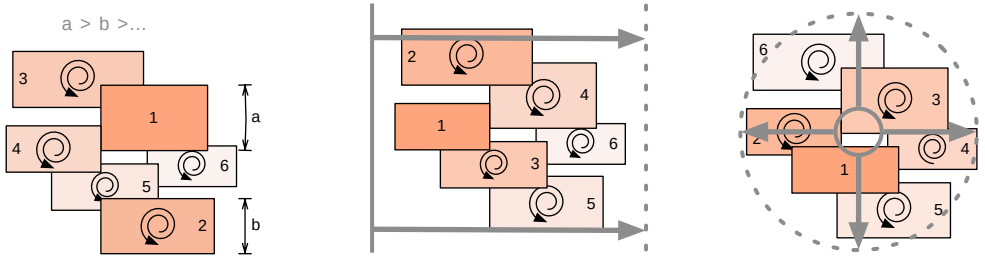
In Information Visualization, a common task is to arrange data points in 2D space. Trivially, data can be intrinsically two-dimensional, i.e. geolocations or layouted text. To observe higher dimensional data, projection algorithms (MDS, PCA, ...) are used to preserve closeness (and distance) of the given data in 2D space. When substituting the data points with dimensional representatives, overlap between these objects can occur. The resulting clutter reduces readability and is therefore unwanted. Removing the overlap is non-trivial and an optimal solution is considered NP-hard [DMS05]. Non-optimal, heuristic solutions are used to achieve reasonably good results. We evaluate common algorithms w.r.t. their performance in minimizing euclidean displacement, retaining orthogonal ordering (described in section 5.2), shape, and size preservation. We developed a structured construction of Wordles that allows a better placement (rolling out the representatives over the 2D plane). The algorithm achieves under constraints of dense positions and non-square items better balanced results than the other methods regarding the mentioned evaluation measures. Throughout the paper we give examples using text labels as representatives although the measures are based on synthetic data with general characteristics.

In the paper, we provide an overview of the related work in the following section. The existing algorithms are described in Section 3. In Section 4 we introduce our algorithm (RWordle) and compare it against the aforementioned methods. The evaluation measures and a discussion of their results can be found in Sections 5 and 6. Application scenarios for our algorithm are given in Section 7. Section 8 concludes and presents future work ideas.

2. Related Work

Multiple graph drawing algorithms address overlap removal either as layout or post-processing step. We want to focus on the latter, because many application scenarios already provide desired positions for the 2D representation of elements (Figure 11 shows a typical infovis application with MDS projection of labels). A first class of algorithms that applies physical models to solve the problem is exemplary represented by Force Scan [MELS95, HLSG07, HIMF98] and Spring Algorithms [FR91, LEN05, CLY04, HK02]. They iteratively solve a global energy function until the overlap is removed. Chuang et.al. and Harel et.al. [CLY04, HK02] in addition focus on non-uniform labels. From computer graphics, physical simulation frameworks [Cat11, Lem11] are known which try to retain the rigid body behavior and therefore solve occlusions of objects. For our comparison, we use Box2D as physical approach, in detail described in Section 3. Gansner et.al. [GH09] use *stress minimization* to a given stress function, dependent on layout crite-

© 2012 The Author(s)
Computer Graphics Forum © 2012 The Eurographics Association and Blackwell Publishing Ltd. Published by Blackwell Publishing, 9600 Garsington Road, Oxford OX4 2DQ, UK and 350 Main Street, Malden, MA 02148, USA.



(a) (Mani-)Wordle (b) RWordle Linear (c) RWordle Concentric

Figure 1: For local overlap evasion methods, different approaches for ordering the elements are shown. ManiWordle (a) uses a size-dependent ordering. We suggest to use a scan-line based linear order (in (b) from left to right) or a concentric order (c).

ria such as proximity or topology, in order to generate the local optimal non-overlapping state out of an initial layout. Another class of overlap removal algorithms is composed of *constrained optimization* algorithms. Dwyer and Marriot [MSTH03, DMS05] generate "separation constraints" that ensure non-overlapping and find an optimized solution via quadratic programming which solves these constraints. Lyons et.al. [Lyo92] apply *Voronoi Cluster busting*, which tries to even the distribution of nodes in areas with high density (clusters). A recent and detailed overview of literature related to graph drawing approaches is given by Gansner and Hu [GH09].

In information visualization, a method for aesthetic layouts of tag clouds [BGN08] has gained popularity. Wordle [VWF09] uses a spiral scheme for the random placement of text labels in order to overcome overlaps. ManiWordle [KLKS10] enriches that idea by adding performance improvements and allow positioning the terms. Kim et.al. [KKEE11] use a space tiling approach for fast approximation of a weight-sorted Wordle layout. The iterative tiling can leave small partitions unsued which decreases packing desnity.

3. Existing Algorithms

The following section describes the algorithms we used for evaluation. Their origins are graph drawing, computer graphics, and information visualization. They are open source which makes them first choices when searching for a practical solution to overlap removal.

3.1. Box2D Rigid Body Physics Engine

Box2D [Cat11] is a widely used open source physics engine written in *C++*. It uses an iterative constraint solver loop [Cat05] for rigid body dynamics. A body is an entity in the physics engine, which has a shape and a position. Overlap is iteratively removed body-to-body wise by using a Baumgarte scheme [Bau72] to push the bodies apart. The velocity of this push is proportional to the penetration depth of the two objects.

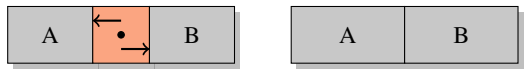


Figure 2: Overlap removal in Box2D: Solving body-to-body constraints by pushing bodies away from each other with applied forces.

The physics engine has the constraint that no bodies can overlap. By representing overlapping objects with bodies in the physics engine, the constraint solver tries to remove existing overlaps.

3.2. VPSC

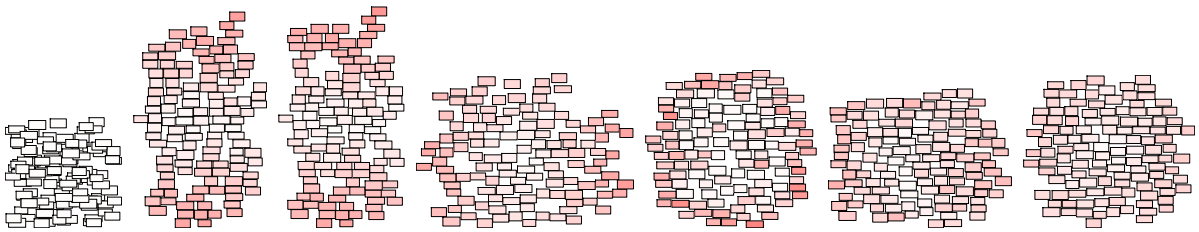
Dwyer et al. [DMS05] propose an algorithm to remove node overlaps in graph layouts, especially focussing on retaining orthogonal ordering. Because we don't have edges in our scenario and the algorithm exclusively operates on graph vertices, we can adapt it for our purposes. This algorithm consists of two parts: The first part generates separation constraints for nodes. The second part tries to find an optimal solution to these constraints.

In the separation generation step, a sweep line along an axis is used to generate non-overlapping constraints in one dimension: When a node is found by the sweep line, its orthogonal neighbors w.r.t. the scan-line direction are computed and two non overlapping constraints (between the found node and each of its neighbors) are generated.

After generating constraints of the form $u + d \leq v$, where d is the minimal gap between the two nodes and u, v are the positions of two nodes, the following constrained optimization problem has to be solved for each dimension, described by Dwyer et al. [DMS05]:

> Variable placement with separation constraints (VPSC): Given n variables $v_1, \ldots, v_n, a$ weight

© 2012 The Author(s)
© 2012 The Eurographics Association and Blackwell Publishing Ltd.



(a) original (b) Box2D (c) VPSC (d) PRISM (e) ManiWordle (f) RWordle-L (g) RWordle-C

Figure 3: Layouts for a synthetic scenario of 120 rectangular representatives. The scenario is shown before overlap removal (a) and after overlap removal (b)–(f) for the described algorithms. The Euclidean Distance for each item is mapped to intensity of red. Box2D and VPSC show a clear stretching along the y axis. PRISM leaves unused white space within the layout. ManiWordle places partially correct but towards the outer bound, displacement increases. Both RWordle approaches distributes the Euclidean Displacement more homogeneously.

$w_i \geq 0$ and a desired value d_i, for each variable and a set of separation constraints C over these variables find an assignment to the variables which minimizes $\sum_{i=1}^{n} w_i \times (v_i - d_i)^2$ subject to C.

The set of separation constraints is then treated as a weighted directed graph called *constraint graph* with the nodes from the former graph and the separation constraints between nodes as the edges. The variables (positions) are then processed in ascending order. Before processing vertex v_i, all previous vertices $v_1, \ldots, v_{i-1}$ already have to satisfy the separation constraints. The already positioned vertices create an already solved "block". Before adding v_i to the block, the separation constraints between v_i and the block have to be solved. With this technique, the algorithm merges nodes into larger and larger blocks until all nodes satisfy the separation constraints. There might however be non-optimal solutions depending for example on the ordering of the nodes.

3.3. PRISM

PRISM tries to remove overlaps while maintaining the proximity relations between the nodes by working on the proximity graph (Delaunay Triangulation). If there is overlap on an edge of this graph, they calculate an overlap factor t_{ij} between the two nodes of this edge.

$$t_{ij} = max\left(min\left(\frac{hw_i + hw_j}{x_i - x_j}, \frac{hh_i + hh_j}{y_i - y_j}\right), 1\right) \quad (1)$$

where hw_i and hh_i denote the half-width and half-height. In case of no overlap $t_{ij} = 1$. If overlap exists, it can be removed by expanding the edge (in other words the distance between the two nodes) by this factor. In order to remove all overlaps, the edges in the proximity graph have to have a greater length d_{ij} close to $t_{ij} \cdot ||x_i - x_j||$, which would result in zero distance between the two objects, accordingly for y.

The main goal is then to minimize the stress function

$$\sum_{(i,j) \in E_P} (||x_i - x_j|| - d_{ij})^2 \quad (2)$$

where E_P denotes the edges of the proximity graph. However, the stress function is not minimized in one step, but in an iterative way to avoid obstructing the original layout by too large displacements of single nodes. Therefore, they damp the overlap factor by setting it to

$$min(t_{ij}, s_{max}) \quad (3)$$

where s_{max} limits the overlap removed in one iteration. After convergence, node overlaps may still be present. A scan-line algorithm is applied to find all overlaps and add them as edges in the proximity graph. The stress function is then re-evaluated and these steps are repeated until the scan-line finds no more overlaps.

3.4. (Mani-)Wordle

Wordle [Wor11] is a web-based visualization tool to generate tag clouds. It aims at generating aesthetic visualizations in terms of typography, color and composition.

For each term, a point is picked randomly, trying to keep the desired overall layout which can be specified in advance. If the term overlaps with any other already positioned term, it is moved in a spiral around its initial position.

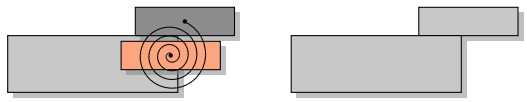


Figure 4: Wordle greedy layout algorithm: In case of overlap, search in a circular manner for a new position.

The only possibility to influence the result is to "play" with the parameters and thus try to achieve the wanted result this way. Koh et al. [KLKS10] try to overcome this weakness

© 2012 The Author(s)
© 2012 The Eurographics Association and Blackwell Publishing Ltd.

by providing more flexible control over Wordle by introducing ManiWordle, a Wordle based tool which allows the user to position terms freely. They also present some speed improvements over the original algorithm.

4. Our Method: RWordle

Physical simulations have an issue with overlap removal of similar, non-square objects. A force in the same direction is applied to every object and the resulting layout includes stacked objects and the overall layout is not compact. A similar problem exists for scan-line approaches like VPSC, because the separation constraints are solved axis-wise. Figure 5 gives a typical example.

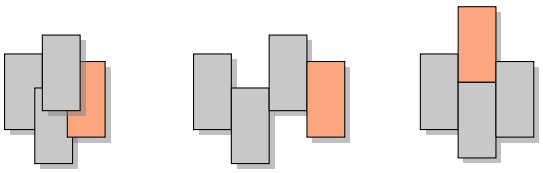


Figure 5: Overlap removal in the same direction may lead to a stacking problem (center) even if another, more compact layout would be preferable (right).

On the other hand (Mani-) Wordle loses orthogonal order because the already layouted representatives (of big size) subdivide the canvas in such a way that the smaller representatives have to search for free space in a bigger radius.

The idea of our algorithm is to improve the orthogonal ordering of Wordle. The new algorithm uses a modified version of Wordle's greedy layout strategy: It adopts the scan-line overlap removal strategy of the *Force Scan* [MELS95] algorithm. In other words, the weight-driven sorting step in the given algorithm for Wordle is substituted by a geometric driven sorting. We propose two different approaches for sorting the elements (see Figure 1):

1. **Linear Sorting (RWordle-L)** The items are sorted along a scan-line which runs along an axis defined by angle α.
2. **Concentric Sorting (RWordle-C)** The items are sorted along their distance to the geometric center of the whole scene.

After sorting the items, the placement follows the same algorithm as ManiWordle, i.e. if overlap occurs an alternative position is searched for in a spiral manner. In the case of linear sorting, the layout extends along the scan-line axes. To lower the effect that this extension destroys orthogonal ordering, the scene of layout items is re-centered to the corresponding center of original positions after each item insertion. The method is summarized in Algorithm 1.

Since the search for good positions only requires an intersection test between polygons, two extensions can easily be implemented for the Wordle-like approaches:

Algorithm 1: The RWorlde layout algorithm.

```
Data: Terms with positions: input
begin
    sort input linear (-L) or circular (-C)
    forall the term, position ∈ input do
        while term overlaps with already layouted
        terms do
            update position on a circle with origin at
            the initial position
        re-center the layouted objects
```

1. **Artificial representative shapes** The intersection test can be applied to any artificial shape and therefore the approaches are not limited to rectangles
2. **Canvas restriction** The intersection with an outer canvas can also be tested, and forbidden areas for label placement can be defined. This is beneficial if the representatives should form a specific outer layout or, in case of clustering, forbidden areas (like an outer canvas restriction) can be defined. However, an abort criterion must be defined to stop the search for positions in case of a full canvas.

For comparison with the other algorithms, we limit the shape of our representatives to axis parallel rectangular bounding boxes.

5. Evaluation Methods

To compare the behavior of the various algorithms we use a set of measures which are derived from Misue et.al. [MELS95]. In this paper the authors state that a mental map is preserved well if a layout fulfills the following conditions:

- **The proximity relations are preserved.** We measure the average *Euclidean Distance* of all labels before and after overlap removal assuming that relations are best preserved if labels can be placed near to their original position.
- **The orthogonal ordering remains similar.** We measure the sum of *pairwise inversions* between labels before and after overlap removal.
- **The topology is preserved.** We assume, that two factors are important to evaluate: *shape and size stability*. The shape of the convex hull given by the original and the overlap-free layout should have the same characteristics, and the increase in layout space should be moderate.

In this section, we give short descriptions of the used measures and test settings. Finally, the obtained results are discussed.

5.1. Euclidean Distance (ED)

We measure the distance between midpoints p_i of a representative in the original layout and the overlap-free layout

© 2012 The Author(s)
© 2012 The Eurographics Association and Blackwell Publishing Ltd.

by calculating the average Euclidean distance:

$$ed = \frac{1}{|P|} \sum_{1 \le i \le |P|} d(p_i, p_i') \tag{4}$$

Where $|P|$ denotes the number of points and $d(p_i, p_i')$ the Euclidean distance between the position p and the new assigned position p'.

5.2. Orthogonal Ordering – No. of Inversions (OO)

The *Orthogonal Ordering* measure (*oo*) describes, in analogy to Misue et. al. [MELS95], changes in the relative positioning of the objects. This measure works on the ordering between two old positions p_i, p_j and the corresponding new positions p_i', p_j'. In two scan line sweeps, one horizontally and one vertically, the orderings of all points before and after the layout adjustment are saved in lists L_x, L_y and L_x', L_y' respectively. The change in the orthogonal ordering is measured as the number of inversions between the vertical and horizontal lists.

$$oo = \sum_{d \in x,y} \sum_{i<j} inv_d^{(t)}(i,j)$$

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

Where i, j are indices of list L and $inv_y^{(t)}(i,j)$ is defined analogous to $inv_x^{(t)}(i,j)$

5.3. Layout Similarity (LS)

Proposed by Gansner et.al. [GH09] we use a variation of the Frobenius metric. First, the Delaunay Triangulation of the original layout is computed. The length of the Delaunay edges before ($l_{DT,i}$) and after overlap removal ($l'_{DT,i}$) are calculated. The normalized standard deviation of all ratios $l'_{DT,i}/l_{DT,i}$ provides a measure which we will call *ls*. For more details we refer to [GH09].

5.4. Shape Preservation (SP)

In addition to the aforementioned measure (*ls*) we introduce a measure *sp* which is independent of the changes in orthogonal order.

The convex hull over all objects in the original (C) and overlap-free layout (C') is calculated. From the corresponding centers of masses (c, c') the lengths $l_{angle} = length(c, p_{C,angle})$ of line segments from the center of mass to the intersection point with the convex hull are calculated for angular orientations from $0° - 350°$ in $10°$ steps. Figure 6 illustrates this. The differences $d_{angle} = l'_{angle}/l_{angle}$ describe the shape increase towards the sampled directions.

We calculate the standard deviation of all difference values:

$$d_{angle} = \frac{l'_{angle}}{l_{angle}} \tag{5}$$

$$sp = \sqrt{\frac{1}{36} \sum_{a=0}^{35} (d_{a \cdot 10°} - \text{mean}(d_x))} \tag{6}$$

If a shape is equally expanded in each direction the standard deviation of the changes in all directions is small because of their similarity, if the shape is heavily deformed the standard deviation is high.

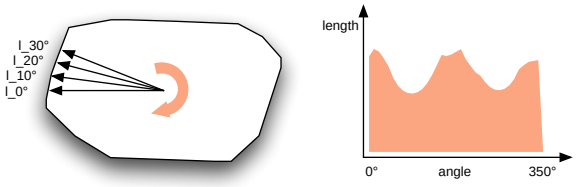


Figure 6: The length of rays from the center of mass of the convex hull intersected with the convex hull are used to describe the shape of the object distribution.

5.5. Size Increase (SI)

Given the convex hulls C and C' (see SP), the size increment is calculated as the ratio:

$$si = \frac{\text{area}(C')}{\text{area}(C)} \tag{7}$$

This way we determine the relative changes in size and thus have a measure of the compactness of the representation.

6. Evaluation Results

In this section, we evaluate the performance of different algorithms on increasing density. Furthermore, in Figure 8 we show for RWordle-L how the direction of the scan-line can modify the resulting layouts and measures. The influence of the relation between inner shape layout and the distribution shape of the representatives is shown in Figure 9. For better clarification of the visual effects using the different layout methods, Figure 3 shows the results of a given test set.

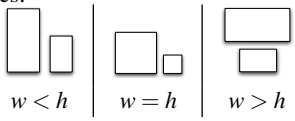
6.1. Test Settings

Setting up test scenarios requires the limitation of degrees of freedom for configuring the shape of representatives and the distribution of 2D positions. The following table gives an overview of degrees of freedom we addressed.

For every test we generate 99 configurations with random assignments of size and position of representatives. After

© 2012 The Author(s)
© 2012 The Eurographics Association and Blackwell Publishing Ltd.

shape of representatives :

parameter	**variations**
form	rectangular, text shape
size	randomly assigned within upper and lower bounds and fixed characteristics: 

shape of distribution of 2D positions :

parameter	**variations**
canvas size	fixed with characteristics: $w > h$, $w = h$, $w < h$ (see shape)
density	controlled by number of 2D positions/objects
point distribution	equally randomly distributed, cluster distribution

Table 1: Degrees of freedom used for evaluation.

each test, the median of each measure is determined independently. This allows us to characterize the general behavior of the algorithms. Although, it is important to keep in mind that the measures are not independent, i.e. the scenario which reflects the median for Euclidean Distance might not be the same scenario for the median of Orthogonal Ordering.

6.2. Density Influence

The density in a synthetic scenario can be controlled by the number of objects in a given canvas. A canvas with the size of 400×400 and an average element area of 160 is used in Figure 7. The previously introduced measures are shown against the number of objects.

For all measures on the left side of Figure 7 using squared representatives, RWordle results are in the middle field. The exception is RWordle-L in Shape Preservation, which retains the outer convex hull much better than the other approaches. PRISM follows the behavior. In the case of non-square representatives ($w > h$), which are shown on the right side of Figure 7, RWordle performs better for high density than the other algorithms with respect to Euclidean Distance, equal to (and best with) Wordle regarding Shape Preservation and Size Increase. In Orthogonal Ordering, it performs significantly better than Wordle.

6.3. Influence of scan-line direction on RWordle-L

In the linear case, RWordle depends on the angle from which the scan-line runs through the scene. Figure 8 shows the results for different angles on a 400×400 canvas with an average element area of 160. Orthogonal Ordering and Shape Preservation measures are influenced by the scan-line direction, but always perform worse than the circular approach. Finding a good scan-line direction in a non brute force manner will be part of future work.

6.4. Correlation between representative shape and distribution shape

The relationship between the shape of representatives and the shape of how these representatives are distributed can influence the avoidance strategy of the different algorithms. Figure 9 shows measures for the same dataset, 120 representatives with an average element area of 160, with a changing initial position distribution: The canvas shape $w < h$ has the opposite aspect ratio to the layouted elements within, whereas the $w > h$ canvas shape has a similar aspect ratio. The Euclidean Distance results for iterative constraint solvers (VPSC, Box2D) depend highly on the distribution of positions, whereas PRISM and the Wordle algorithms are more stable w.r.t. these changes. Measuring Shape Preservation, all algorithms depend on the constellation between representative shape and distribution shape with RWordle-C performing best.

6.5. Runtime and Scalability

VPSC and PRISM are optimized for performance. Our approach and Box2D are a magnitude slower than these, but terminates for the given scenarios within a reasonable short time (less than a second). While the other algorithms guarantee overlap removal, Box2D can fail if the number of simulation steps is too low. The main cost for RWordle is the occlusion test with other objects. It will benefit from an optimized data structure like a quad tree. Under this precondition RWordle will allow scalability to very large scenarios.

7. Applications

We apply RWordle on two real world data examples, one from a geographical and one from a biological domain.

7.1. Geolocated Data

We applied the approaches to a map of Great Britain, positioning each city name label in the center of the city's geographic location and mapping the number of inhabitants logarithmically to font size. The resulting images are given in Figure 10. The layouts clearly show the mentioned drawbacks and benefits of the discussed algorithms. VPSC (and Box2D) stretch the layout along the y-axis. To avoid this one dimensional stretching, PRISM focus on minimizing stress while using both dimensions, and therefore expanding in both. ManiWorlde has to extend the layout eastwards, because the huge terms already fragment the space hindering the placement of smaller items. RWordle-C retains the outer shape well. In addition an example for RWordle-C is given

© 2012 The Author(s)
© 2012 The Eurographics Association and Blackwell Publishing Ltd.

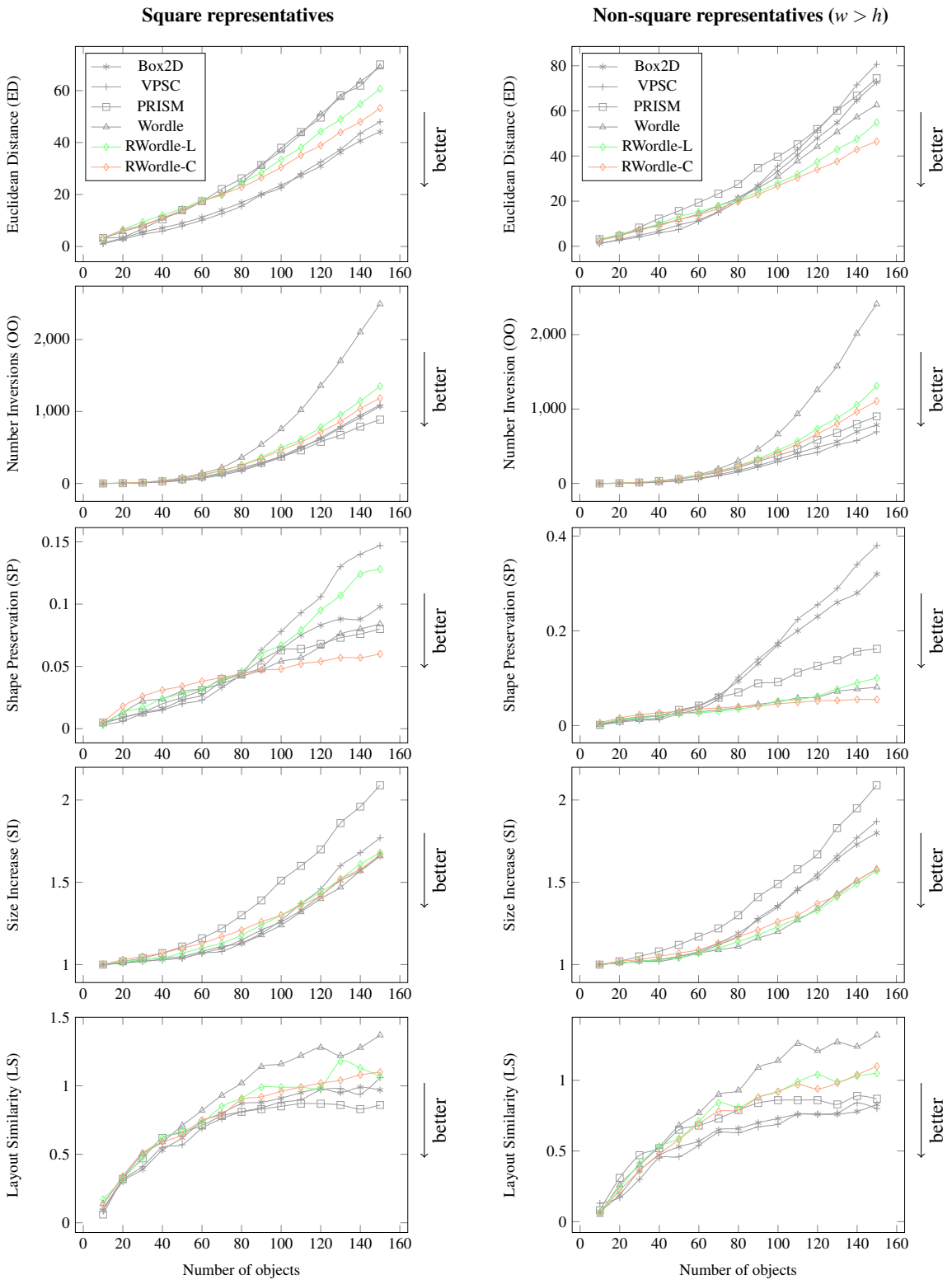


Figure 7: Detailed measures for the described algorithms with increasing density. On the left hand side for square representatives, on the right hand side for non-square representatives ($w > h$). A canvas with the size of 400×400 and an average element area of 160 is used.

© 2012 The Author(s)
© 2012 The Eurographics Association and Blackwell Publishing Ltd.

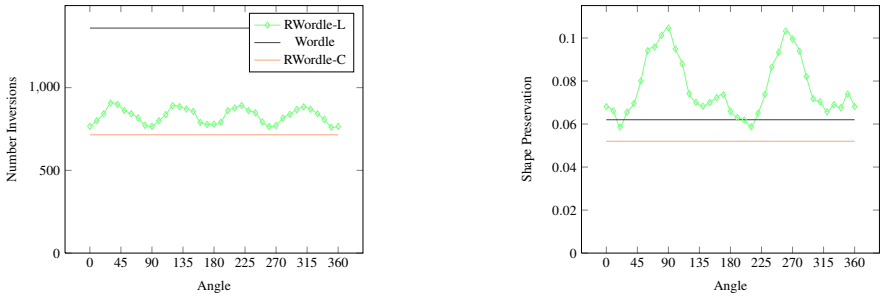
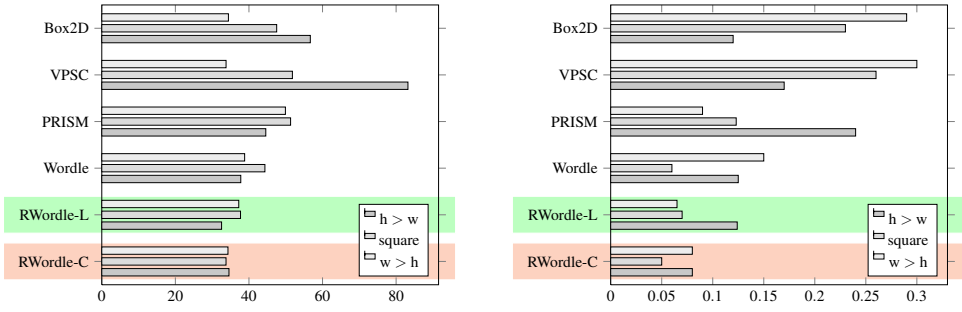


Figure 8: The influence of the angle of the scan-line direction on the linear approach (RWordle-L). Both measures correlate with the angle, but are not as good as for the concentric approach (RWordle-C).



(a) Euclidean Distance (ED) (b) Shape Preservation (SP)

Figure 9: The shape ($w > h$, square, $w < h$) of initial distribution of non-square representatives affects the Euclidean Distance measure for VPSC and Box2D heavily. RWordle is more stable w.r.t. this measure and delivers good results for ED and SP.

with occlusion tests in the text shapes instead of bounding boxes.

7.2. Projected Data

Another example for applying the overlap removal is given in Figure 11. For investigating biochemical molecules of a High-Throughput Screening experiment ([BSB*11]), representatives of molecule structure and additional informations (e.g. ID, water solubility) are mapped into the 2D plane. The structural difference between molecules is calculated as the Tanimoto distance [Tan58] of structural fingerprint bitvectors. These distances are input to an MDS projection. The positions where the labels should occur are given in the upper left corner of Figure 11. The overlap free layout given by our method shows that separated clusters remain well separated, the layout is compact and the outer shape is preserved. On the other hand, the moderate loss in retaining the Orthogonal Ordering can be seen in the lower right part where clusters intersect.

8. Conclusion and Future Work

We provided an in-depth investigation of state-of-the-art (and free available) overlap removal algorithms and contribute a new algorithm stemming from the Wordle idea. VPSC and Box2D (as representative of physic based overlap removal) retain orthogonal order very well, however the disadvantages of the stacking problem have to be taken into consideration here as well as a significant distortion of the layout with regard to dense cases. PRISM copes with this problem and focuses on a trade-off between orthogonal ordering and layout preservation for the cost of more space and worse outer Shape Preservation. RWordle-C has clear benefits in retaining the outer shape and providing a compact layout, while keeping the difference in Orthogonal Ordering moderate. It is especially useful in dense cases with non-square data representatives.

For future work, we want to determine a good scan-line direction for the RWordle-L algorithm, which can also be beneficial for other scan-line based approaches. Another interesting problem is a scenario where the labels are distributed along a diagonal. All of the discussed algorithms did not perform well in this case. The use of our algorithms as a cluster busting method and the later reapplication on the formed clusters is another idea for future work.

© 2012 The Author(s)
© 2012 The Eurographics Association and Blackwell Publishing Ltd.

(a) original (b) VPSC (c) PRISM

(d) ManiWordle (e) RWordle-C (f) RWordle-C with text shapes

Figure 10: The different overlap removal algorithms applied to the map of England. The number of inhabitants is logarithmically mapped to the font size. (f) shows overlaps removed with text shapes instead of bounding boxes.

Figure 11: Representatives of molecule structure and additional informations (e.g. ID, water solubility) are mapped into the 2D plane according to their structural similarity. We applied our algorithm RWordle-C to remove overlap.

© 2012 The Author(s)
© 2012 The Eurographics Association and Blackwell Publishing Ltd.

References

[Bau72] BAUMGARTE J.: Stabilization of constraints and integrals of motion in dynamical systems. *Computer Methods in Applied Mechanics and Engineering 1*, 1 (1972), 1–16. 2

[BGN08] BATEMAN S., GUTWIN C., NACENTA M.: Seeing things in the clouds: The effect of visual features on tag cloud selections. In *Proceedings of the ACM Conference on Hypertext and Hypermedia (Hypertext '08)* (Pittsburgh, US, 2008), pp. 193–202. 2

[BSB*11] BERTINI E., STROBELT H., BRAUN J., DEUSSEN O., GROTH U., MAYER T. U., MERHOF D.: Hitsee: A visualization tool for hit selection and analysis in high-throughput screening experiments. In *Proceedings of 1st IEEE Symposium on Biological Data Visualization (IEEE BioVis)* (2011), pp. 95–102. 8

[Cat05] CATTO E.: *Iterative dynamics with temporal coherence*. 2005, pp. 1–24. 2

[Cat11] CATTO E.: Box2d - a 2d physics engine for games. http://box2d.org, September 2011. 1, 2

[CLY04] CHUANG J.-H., LIN C.-C., YEN H.-C.: Drawing graphs with nonuniform nodes using potential fields. In *Graph Drawing*, Liotta G., (Ed.), vol. 2912 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2004, pp. 460–465. 1

[DMS05] DWYER T., MARRIOTT K., STUCKEY P. J.: Fast node overlap removal. In *In: Proc. 13th Int. Symp. on Graph Drawing (GD'05). Volume 3843 of LNCS. (2006) 153–164* (2005), Springer, pp. 153–164. 1, 2

[FR91] FRUCHTERMAN T. M. J., REINGOLD E. M.: Graph drawing by force-directed placement. *Software: Practice and Experience 21*, 11 (1991), 1129–1164. 1

[GH09] GANSNER E. R., HU Y.: Graph drawing. Springer-Verlag, Berlin, Heidelberg, 2009, ch. Efficient Node Overlap Removal Using a Proximity Stress Model, pp. 206–217. 1, 2, 5

[HIMF98] HAYASHI K., INOUE M., MASUZAWA T., FUJIWARA H.: A layout adjustment problem for disjoint rectangles preserving orthogonal order. In *Proceedings of the 6th International Symposium on Graph Drawing* (London, UK, 1998), GD '98, Springer-Verlag, pp. 183–197. 1

[HK02] HAREL D., KOREN Y.: Drawing graphs with nonuniform vertices. In *Proceedings of the Working Conference on Advanced Visual Interfaces* (New York, NY, USA, 2002), AVI '02, ACM, pp. 157–166. 1

[HLSG07] HUANG X., LAI W., SAJEEV A. S. M., GAO J.: A new algorithm for removing node overlapping in graph visualization. *Inf. Sci. 177* (July 2007), 2821–2844. 1

[KKEE11] KIM K., KO S., ELMQVIST N., EBERT D. S.: Wordbridge: Using composite tag clouds in node-link diagrams for visualizing content and relations in text corpora. In *HICSS'2011* (2011), pp. 1–8. 2

[KLKS10] KOH K., LEE B., KIM B., SEO J.: Maniwordle: providing flexible control over wordle. *IEEE Transactions on Visualization and Computer Graphics 16*, 6 (2010), 1190–1197. 2, 3

[Lem11] LEMBCKE S.: Chipmunk physics. http://chipmunk-physics.net/, September 2011. 1

[LEN05] LI W., EADES P., NIKOLOV N.: Using spring algorithms to remove node overlapping, 2005. 1

[Lyo92] LYONS K. A.: Cluster busting in anchored graph drawing. In *Proceedings of the 1992 conference of the Centre for Advanced Studies on Collaborative research - Volume 1* (1992), CASCON '92, IBM Press, pp. 7–17. 2

[MELS95] MISUE K., EADES P., LAI W., SUGIYAMA K.: Layout Adjustment and the Mental Map. *Journal of Visual Languages & Computing 6*, 2 (June 1995), 183–210. 1, 4, 5

[MSTH03] MARRIOTT K., STUCKEY P., TAM V., HE W.: Removing node overlapping in graph layout using constrained optimization. *Constraints 8* (April 2003), 143–171. 2

[Tan58] TANIMOTO T.: *An elementary mathematical theory of classification and prediction*. International Business Machines Corporation, 1958. 8

[VWF09] VIEGAS F. B., WATTENBERG M., FEINBERG J.: Participatory visualization with wordle. *IEEE Transactions on Visualization and Computer Graphics 15* (November 2009), 1137–1144. 2

[Wor11] Wordle. http://www.wordle.net, September 2011. 3

© 2012 The Author(s)
© 2012 The Eurographics Association and Blackwell Publishing Ltd.