Uniform Sample Distribution in Scatterplots via Sector-based Transformation

Hennes Rave[*]
University of Münster, Germany

Vladimir Molchanov[†]
University of Münster, Germany

Lars Linsen[‡]
University of Münster, Germany

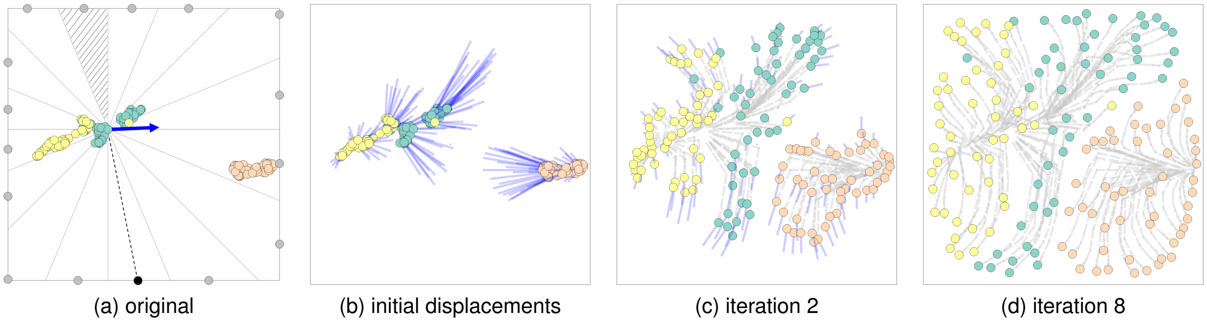
(a) original (b) initial displacements (c) iteration 2 (d) iteration 8

Figure 1: Sector-based transformation of a UMAP embedding of the Iris dataset [1]. (a) 16 sectors and anchor points for a selected sample are shown for the original scatterplot. The black anchor point at the bottom belongs to the highlighted sector at the top. Samples are moved toward a sector's anchor point based on the point density inside that sector. The resulting displacement vector is shown in blue. (b) Initial displacement vectors for all samples. (c) Displacement vectors and trajectories of samples after two iterations of the proposed algorithm. (d) Sample distribution and deformation trajectories after 8 iterations.

Abstract

A high number of samples often leads to occlusion in scatterplots, which hinders data perception and analysis. De-cluttering approaches based on spatial transformation reduce visual clutter by remapping samples using the entire available scatterplot domain. Such regularized scatterplots may still be used for data analysis tasks, if the spatial transformation is smooth and preserves the original neighborhood relations of samples. Recently, Rave et al. [21] proposed an efficient regularization method based on integral images. We propose a generalization of their regularization scheme using sector-based transformations with the aim of increasing sample uniformity of the resulting scatterplot. We document the improvement of our approach using various uniformity measures.

Index Terms: Scatterplot de-cluttering, spatial transformation.

1 Introduction

Bi-variate discrete data can be presented in a scatterplot as a set of points. Variations and correlations of the points' positions, density, and inter-sample distances reflect important characteristics of the data. However, high local density of samples in scatterplots can lead to occlusion and visual clutter. Cluttered regions are difficult to perceive, explore, compare, and characterize. Therefore, various de-cluttering methods have been proposed over the last decades. One approach for mitigating occlusion is to relax the positions of samples in the scatterplot towards a uniform sample distribution, which more effectively uses the available visual space. When not handled with care, such a transformation of the scatterplot domain may significantly change the structure of the visually represented data, which would make any data analysis meaningless. Therefore, the construction of the regularizing mapping needs to preserve certain properties of the scatterplot.

Recently, Rave et al. [21] proposed a de-cluttering algorithm, which iteratively changes the positions of the samples based on information about their global distribution. This information is encoded in a set of *integral images* (InIms). The authors have shown that globally smooth transformations of the visual domain approximately preserve the initial ordering of the samples as well as several other significant quality measures. Moreover, the proposed algorithm has lower complexity and performs one order of magnitude better in terms of execution time than competing approaches. Thus, large datasets can be regularized at interactive rates using the proposed algorithm.

InIms can be efficiently computed on the GPU. To compute InIms, the visual domain is systematically split into four non-overlapping quadrants. Then, the number of samples in each quadrant is computed. The disbalance of the number of samples relative to the areas of the quadrants determines a local displacement needed to correct the disbalance. However, splitting the domain into a higher number of subdomains may increase the spatial resolution, provide more precise information about global sample distribution, and result in a better regularizing transformation.

We propose a generalization of the method by Rave et al. [21] by increasing its angular resolution, i.e., replacing InIms quadrants with a larger number of sectors that partition the visual domain. We further propose to operate on discrete samples, which eliminates the need for a regularization radius as a parameter of the algorithm. We demonstrate that the higher number of sectors increases the uniformity of the spatial arrangements of the samples in the resulting scatterplot. We evaluate various quality measures, discuss the interplay of the parameters, and visualize the resulting transformation.

2 Related Work

Ellis and Dix [9] identified three main groups of techniques for **reducing clutter** in scatterplots. The first group of methods affects the appearance of the samples in the plot by adjusting their size [27]

[*] e-mail: hennes.rave@uni-muenster.de
[†] e-mail: molchano@uni-muenster.de
[‡] e-mail: linsen@uni-muenster.de

or opacity [10, 15], data down-sampling [6, 28], aggregation [16] and clustering [19]. The second group of techniques uses animations to reveal all data samples to the observer at different times. For instance, Chen et al. [2] suggested using flickering points in multiclass scatterplots. Finally, the third group of methods uses a spatial transformation to remap the samples over the visual domain aiming at a more even sample distribution. Using spatial transformation has the advantage that it facilitates (up to some extent) the simultaneous display of all individual samples, which is important for interactions such as selection or brushing.

Following the idea of de-cluttering by **spatial transformation**, Keim et al. [12] let the user control and balance the degree of overlap and distortion in generalized scatterplots. Raidou et al. [20] arranged samples on a rasterized canvas and used color to encode local distortions. Cutura et al. [4, 5] mapped samples to the centers of a grid using a space-filling curve. Hilasaca et al. [11] removed overlaps of the glyphs and evaluated the proposed method using several quality measures. The taxonomy by Ward [26] focuses particularly on displacement methods and emphasizes the importance of user control on the distortion degree and smoothness of the transformation.

Most recent transformation-based de-cluttering techniques rearrange samples in scatterplots on an auxiliary grid, e.g., [20, 5, 11]. While this may help to remove the overlap of glyphs, gridded ordering introduces artificial patterns and alignments when samples are shown as dots. Thus, we advocate for a *continuous transformation* of the visual space.

Often, occlusion is resolved locally, i.e., samples from overpopulated regions are redistributed over neighboring free space. However, if locally available free space is insufficient, existing algorithms like Hagrid [5] have to resolve the collisions. Resolving collisions not only negatively impacts the computation time but also distorts original data ordering. Thus, we conclude that transformations considering data distributions *globally* are preferable.

Finally, algorithms based on general error functional optimization (e.g., Pixel-Relaxed Scatter Plots [20] and DGrid [11]) usually have high complexity and scale poorly with the number of samples. However, occlusion becomes an issue as the number of samples increases. Therefore, the application of optimization algorithms for large datasets is hardly possible, especially if *interactivity* is required.

The de-cluttering method recently presented by Rave et al. [21] overcomes the listed drawbacks of other approaches using **InIms**. It iteratively moves samples according to their current global distribution. At each iteration, discrete samples are convolved with a regularization kernel of proper scale and summed up to obtain a rasterized density distribution. Then, this density is encoded in a system of InIms, which determine local samples' displacements. The method allows for de-cluttering datasets with millions of samples at interactive rates. Thus, the approach by Rave et al. allows for continuous transformation considering global distributions at interactive rates. We generalize the idea of Rave et al., inheriting all its desirable properties, but improve the uniformity of the scatterplot's sample distribution.

3 De-cluttering Using Integral Images

InIms, also called *summed-area tables*, were introduced by Crow [3]. Viola and Jones [25] used InIms in image analysis for object detection. Tilted InIms differ from classical InIms by rotating the summation areas by 45°. Lienhart et al. [14, 13] presented an efficient computation of tilted InIms on the CPU. Molchanov and Linsen [18] used InIms and tilted InIms for map deformation. Approximate InIms can be computed at arbitrary angles as studied by Chin et al. [23].

The de-cluttering technique proposed by Rave et al. [21] exploits InIms and tilted InIms to construct a smooth transformation based on the global sample distribution that can be expressed by an explicit formula. Discrete samples are first convolved with a regularization kernel of radius r resulting in a density distribution d_r. The density is rasterized on a texture of size $2^k \times 2^k$ and a positive constant d_0 is added to all pixel values to avoid singularities. Then, sets of InIms $\{\alpha, \beta, \gamma, \delta\}$ and tilted InIms $\{\alpha_t, \beta_t, \gamma_t, \delta_t\}$ are computed by summing up values of texture $d = d_r + d_0$ over subdomains depicted in Fig. 2. Finally, the de-cluttering transformation can be computed at each pixel location $(i,\, j)$ using normalized coordinates $(x,\, y) = 2^{-k}\,(i,\, j)$ according to the formula:

$$t(x,\, y) = (x,\, y) + t(x,\, y;\, d) - t(x,\, y;\, d_0), \tag{1}$$

where

$$\begin{aligned} t(x,\, y;\, d) = \big(\alpha \cdot q_1(x,\, y) \quad &+ \beta \cdot q_2(x,\, y) \quad &+ \\ \gamma \cdot q_3(x,\, y) \quad &+ \delta \cdot q_4(x,\, y) \quad &+ \\ \alpha_t \cdot (x,\, 1) \quad &+ \beta_t \cdot (1,\, y) \quad &+ \\ \gamma_t \cdot (x,\, 0) \quad &+ \delta_t \cdot (0,\, y)\;\big)\big/(2C) \end{aligned} \tag{2}$$

Equation 2 weights the eight anchor points with the values of the InIms and divides by the total mass $C = \sum d(i,j)$, where anchor points q_1, q_2, q_3, q_4 are shown in Fig. 2. The displacement map t can be computed at the samples' locations using bi-linear interpolation. After changing the positions of all samples according to the interpolated t, the density d can be re-computed for the next iteration. Thus, the samples are pulled towards the respective anchor points based on the global sample distribution. The iterative mapping stops when the sample distribution becomes nearly uniform.

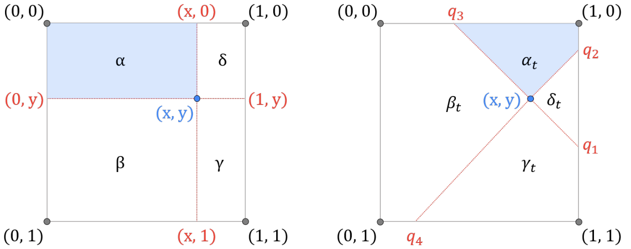

Figure 2: *Left:* The four InIm coefficients are computed at location (x,y) by summing up a density function over respective rectangular regions. *Right:* The four tilted InIms are computed for the same location by summing up the density over tilted regions. (Image reproduced from Molchanov and Linsen [18].)

4 Sector-based Transformation

The deformation vector $t(x,y;d)$ in Eq. (2) is computed by multiplying classical InIms and tilted InIms with respective anchor points and summing them up. When considering the first four and the last four terms in Eq. (2) separately, one can relate them to left and right schemes in Fig. 2, correspondingly. In each case, the scatterplot domain is partitioned into four non-overlapping right-angled sectors around position (x,y). Then, values of InIms represent cumulative mass in each of the sectors. Equation (1) detects the imbalance of the sample distribution around (x,y) by comparing $t(x,y;d)$ with its counterpart computed for ideal uniform density distribution d_0.

Splitting the domain into four sectors is dictated by using an efficient algorithm for computing (tilted) InIms on the GPU by Rave et al. [21]. However, an important research question is left open: How does increased angular resolution (i.e., splitting the domain into a larger number of sectors) affect the rate of the regularization, uniformity, and other quality metrics of the resulting layout? Our work answers these questions by proposing a sector-based transformation and performing respective numerical experiments.

Furthermore, the de-cluttering algorithm by Rave et al. depends on three parameters: regularization radius r, texture size exponent k, and additive constant density d_0. The authors performed an extensive study on the effect of varying r and d_0 and gave explicit recommendations for choosing values of these parameters. The choice of k mainly depends on the available hardware characteristics. In our work, we propose to operate directly with the original discrete distribution of samples in the scatterplot, thus eliminating the need for all three parameters, i.e., regularization radius r, the additive density d_0, and the texture size exponent k.

Our proposed algorithm performs the following steps: First, we split the domain into m sectors $\mathscr{S}_i(\mathbf{x}_u)$ of equal angles around each sample $\mathbf{x}_u = (x_u, y_u)$ in the scatterplot. Then, we count the number of samples $s_i(\mathbf{x}_u)$ in each sector $\mathscr{S}_i(\mathbf{x}_u)$. Each sector's centerline intersects the boundary of the scatterplot domain at two points. Of those two, the intersection point which does not lie inside the sector is chosen as the anchor point $q_i(\mathbf{x}_u)$, see Fig. 1(a). Then, the image of sample $\mathbf{x}_u$ at each iteration can be found according to the following formula:

$$t(\mathbf{x}_u) = \mathbf{x}_u + \sum_i q_i(\mathbf{x}_u) \cdot \left(\frac{s_i(\mathbf{x}_u)}{n} - \frac{|\mathscr{S}_i(\mathbf{x}_u)|}{|\mathscr{S}|} \right), \tag{3}$$

where $|\mathscr{S}_i|$ is the area of sector $\mathscr{S}_i$ and $|\mathscr{S}|$ is the total area of the scatterplot domain. The iterations stop when the required level of uniformity of the sample distribution or the maximal number of iterations is reached, depending on the application environment and the user preferences. The proposed de-cluttering procedure allows for a smooth transition between the initial and fully regularized layouts to maintain a mental map of the performed deformation.

5 Results

We compare our sector-based transformation approach to the InIms-based de-cluttering algorithm by Rave et al. [21]. The quality measures presented in Fig. 4 and in Fig. 7 were computed for the same 2,832 scatterplot layouts from the UCI repository [8] used by Rave et al. [21]. We used the suggested combination of the algorithm parameters, namely $r = 8$, $k = 10$, and $d_0 = n/(2^k \times 2^k)$ for the InIms-based de-cluttering.

Uniformity. Improving the uniformity of the resulting scatterplot layout was the main goal of our approach. The visual evaluation of the sample distribution shows that our regularization algorithm using 64 sectors results in a more even sample distribution of an artificial dataset than the InIms-based approach, see Fig. 3. Corners of the rectangular domain represent singularities affecting the sample distribution in their neighborhoods. Mitigating the effects of non-smooth domain boundary can be a direction of future work.

To quantify and compare the regularity of the samples' positions, we first apply the same procedure as Rave et al.: We split the scatterplot domain into bins of size 4×4 pixels, find the number of samples in each of these bins, and compute the standard deviation of these values from the mean number of samples per bin. For a perfectly uniform distribution of samples, the standard deviation should vanish.

Another quality measure of samples' uniformity is the Ripley function [22]

$$L^2(y) = \frac{\|\mathscr{S}\|}{n^2 \pi} \sum_{u \neq v} w_{uv} \cdot \mathrm{I}(\|\mathbf{x}_u - \mathbf{x}_v)\| < y), \tag{4}$$

where w_{uv} is a correction coefficient due to boundary effects [7] and $\mathrm{I}(y)$ is the indicator (or characteristic) function. When $L(y)$ is evaluated for values of y between zero and half of the domain's shortest dimension (value 0.5 in scatterplot coordinates), deviation from uniformity of the sample distribution can be evaluated as

$$\int_0^{0.5} |L(y) - y| \,\mathrm{d}y \approx \frac{1}{10} \sum_{i=1,\ldots,10} |L(0.05 \cdot i) - 0.05 \cdot i|,$$

where we approximate the integral by computing $L(y) - y$ for 10 evenly spaced values of y between 0.05 and 0.5.

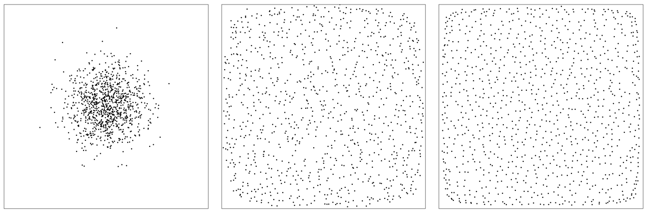
(a) original (b) InIms-based (c) sector-based

Figure 3: De-cluttering scatterplot with artificial data (a) using the InIms-based approach by Rave et al. [21] (b) and using our sector-based regularization with 64 sectors (c). Our approach demonstrates a more even distribution of samples and mitigated boundary effects. Future work may address the distribution artifacts near the corners of the boundary.

Results of the uniformity estimations based on the standard deviation of the binned number of samples and the Ripley function are presented in Fig. 4. The paired Wilcoxon test shows that our method is significantly better for both measures.

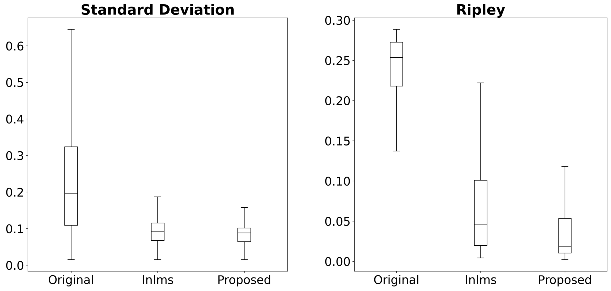

Figure 4: Comparison of uniformity measures for InIms-based and our sector-based regularization methods. The improvements in both quality measures are statistically significant. For context, we also provide the measures for the original scatterplot. Datasets from the UCI repository were used.

Sparsity. The uniformity of the sample distribution can be related to the area of free space around each sample. In a perfectly uniform scatterplot, these values should be close to $|\mathscr{S}|/n$ for all samples. We propose using *sparsity* as another uniformity measure for comparing the scatterplots' layouts. For each sample $\mathbf{x}_u$, we find an associated radius R_u as the minimum of the sample distance to the domain boundary and half its distance to the nearest neighbor. Using the hexagon-packing model, we compute the area of the sample-centered hexagon as $h_u = 3.46 \cdot R_u^2$. Note that these sample-centered hexagons do not overlap and entirely belong to the scatterplot domain. Then, we sum up the areas of all hexagons and relate the sum to the total domain area by computing

$$\text{sparsity} = \sum_u \frac{h_u}{|\mathscr{S}|}.$$

Sparsity values close to unity correspond to a highly uniform sample distribution since the total area of non-overlapping hexagons is close to the domain area.

Figure 5 shows how the sparsity depends on the number of iterations (left) and the number of sectors (right). Increasing the number of iterations for a fixed number of sectors (512) and a fixed number of points (4,096) results in a monotonic increase of sparsity. In particular, the sparsity linearly grows when doubling the number of iterations. For a fixed number of iterations (8), sparsity increases with the number of sectors up to 64 sectors and then remains stable. We observe that datasets with a lower number of points show a stronger sparsity increase. Synthetic datasets used for these numerical tests can be found in the supplementary material.

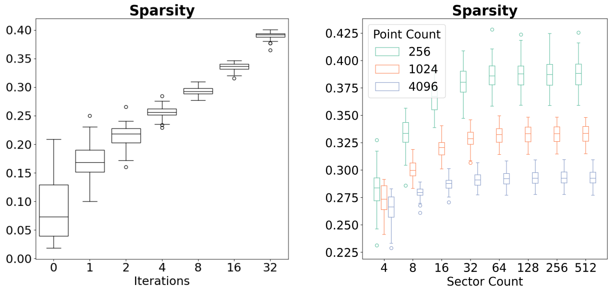

Figure 5: Sparsity values for scatterplots of synthetic datasets. *Left:* Sparsity increases monotonically with the number of iterations (number of points 4,096 and number of sectors 512 are fixed). *Right:* The positive effect of increasing number of sectors is stronger for datasets with fewer points (number of iterations 8 is fixed).

Overplotting. Overplotting can be computed as the difference between the total number of samples and the number of occupied pixels, divided by the total number of samples [21]. The results of the numerical tests for synthetic datasets are shown in Fig. 6. The overplotting rapidly vanishes when iteratively applying the regularization mapping for a fixed number of samples and sectors. When increasing the number of sectors, the maximal value of the overplotting over all datasets approaches zero.

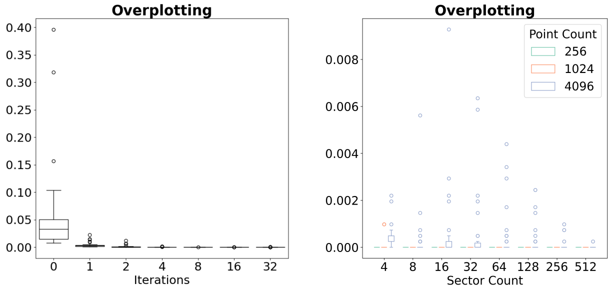

Figure 6: Overplotting values for scatterplots for synthetic datasets. *Left:* Overplotting rapidly vanishes with increasing the number of iterations (number of points 4,096 and number of sectors 512 are fixed). *Right:* Overplotting decreases (even for outliers) with increasing number of sectors (number of iterations 8 is fixed).

Ordering and Trustworthiness. Preserving the neighborhood relations of the original samples' in the regularized layout is crucial. Some quantitative measures of local samples' structure are *trustworthiness* [24] and *orthogonal ordering* [17]. Results of the numerical tests are presented in Fig. 7. Both quality measures were used by Hilasaca et al. [11] and Rave et al. [21] to evaluate the results. For the proposed sector-based approach, these measures demonstrate a statistically significant decrease, i.e., ordering and trustworthiness are better preserved by the InIms-based method.

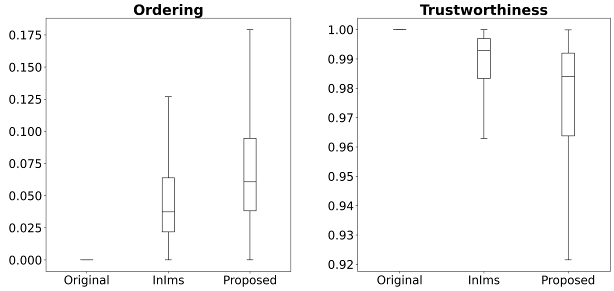

Figure 7: *Left:* Preservation of ordering in the layouts computed by InIms- and our sector-based methods. *Right:* Trustworthiness of the resulting de-cluttered layouts for the two methods when compared to the initial configuration. Datasets from the UCI repository were used.

We note that ordering can be changed by a solid rotation of the original layout with no changes in relative samples' positions. However, such transformation has no effect on the represented data structures and therefore does not constitute an issue for data analysis tasks despite a decrease of the ordering value. The observed decrease in the trustworthiness measure after regularization can be partially explained by the reduction of spatial gaps between clusters. So, samples originally located at the boundary of a cluster become neighbors of the boundary samples of other clusters. This side effect of the regularization procedure does not influence the data analysis provided original clusters can be identified in the regularized layout. Rave et al. [21] thoroughly discussed several approaches for visual encoding of original cluster structures after decluttering. Therefore, an insignificant decrease in the trustworthiness measure should be considered acceptable. Thus, we argue that ordering and trustworthiness are no proper descriptive quality measures for de-cluttering algorithms that use spatial transformations.

6 Conclusion and Discussion

We proposed a sector-based transformation as a generalization to the state-of-the-art de-cluttering algorithm by Rave et al.[21]. Our numerical experiments showed that using our sector-based regularization scheme with a sufficiently large number of sectors significantly improves the uniformity of the resulting scatterplot. The computation times for our approach are higher when compared to Rave et al., but for a fair comparison we would need to optimize our approach including a GPU implementation, which we leave for future work. The current implementation reaches interactive frame rates for up to a few thousand points.

The proposed generalization directly operates with discrete samples. Thus, the three parameters of the original method – the regularization radius r, the texture size exponent k, and the additive constant density d_0 – are no longer needed. Our algorithm only requires the specification of the number of sectors. We analyzed the behavior with an increasing number of sectors and believe that a recommended value for this parameter can be determined once the computation time of the algorithm has been optimized in future work.

Acknowledgments

This work was funded by the Deutsche Forschungsgemeinschaft (DFG) – MO 3050/2-3 – 360330772 and CRC 1450 – 431460824.

References

[1] E. Anderson. The species problem in iris. *Annals of the Missouri Botanical Garden*, 23(3):457–509, 1936. doi: 10.2307/2394164 1

[2] H. Chen, S. Engle, A. Joshi, E. D. Ragan, B. F. Yuksel, and L. Harrison. Using animation to alleviate overdraw in multiclass scatterplot matrices. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pp. 1—12. Association for Computing Machinery, New York, NY, USA, 2018. doi: 10.1145/3173574.3173991 2

[3] F. C. Crow. Summed-area tables for texture mapping. In *Proceedings of the 11th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 207–212. Association for Computing Machinery, New York, NY, USA, 1984. doi: 10.1145/800031.808600 2

[4] R. Cutura, C. Morariu, Z. Cheng, Y. Wang, D. Weiskopf, and M. Sedlmair. Hagrid – gridify scatterplots with Hilbert and Gosper curves. In *Proceedings of the 14th International Symposium on Visual Information Communication and Interaction*, VINCI '21. Association for Computing Machinery, New York, NY, USA, 2021. doi: 10.1145/3481549.3481569 2

[5] R. Cutura, C. Morariu, Z. Cheng, Y. Wang, D. Weiskopf, and M. Sedlmair. Hagrid: using Hilbert and Gosper curves to gridify scatterplots. *Journal of Visualization*, 25:1291—1307, 2022. doi: 10.1007/s12650-022-00854-7 2

[6] A. Dix and G. Ellis. By chance enhancing interaction with large data sets through statistical sampling. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, AVI '02, pp. 167–176. Association for Computing Machinery, New York, NY, USA, 2002. doi: 10.1145/1556262.1556289 2

[7] P. M. Dixon. *Ripley's K Function*. John Wiley & Sons, Ltd, 2006. doi: 10.1002/9780470057339.var046 3

[8] D. Dua and C. Graff. UCI Machine Learning Repository, 2019. 3

[9] G. Ellis and A. Dix. A taxonomy of clutter reduction for information visualisation. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1216–1223, November 2007. doi: 10.1109/TVCG.2007.70535 1

[10] J.-D. Fekete and C. Plaisant. Interactive information visualization of a million items. In B. B. Bederson and B. Shneiderman, eds., *The Craft of Information Visualization*, Interactive Technologies, pp. 279–286. Morgan Kaufmann, San Francisco, 2003. doi: 10.1016/B978-155860915-0/50034-2 2

[11] G. M. Hilasaca, W. E. Marcílio-Jr, D. M. Eler, R. M. Martins, and F. V. Paulovich. A grid-based method for removing overlaps of dimensionality reduction scatterplot layouts. *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–14, 2023. doi: 10.1109/TVCG.2023.3309941 2, 4

[12] D. A. Keim, M. C. Hao, U. Dayal, H. Janetzko, and P. Bak. Generalized scatter plots. *Information Visualization*, 9(4):301–311, 2010. doi: 10.1057/ivs.2009.34 2

[13] R. Lienhart, A. Kuranov, and V. Pisarevsky. Empirical analysis of detection cascades of boosted classifiers for rapid object detection. In B. Michaelis and G. Krell, eds., *Pattern Recognition*, vol. 2781, pp. 297–304. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003. doi: 10.1007/978-3-540-45243-0_39 2

[14] R. Lienhart and J. Maydt. An extended set of Haar-like features for rapid object detection. In *Proceedings. International Conference on Image Processing*, vol. 1, pp. 900–903, Sep 2002. doi: 10.1109/ICIP.2002.1038171 2

[15] J. Matejka, F. Anderson, and G. Fitzmaurice. Dynamic opacity optimization for scatter plots. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pp. 2707—2710. Association for Computing Machinery, New York, NY, USA, 2015. doi: 10.1145/2702123.2702585 2

[16] A. Mayorga and M. Gleicher. Splatterplots: Overcoming overdraw in scatter plots. *IEEE Transactions on Visualization and Computer Graphics*, 19(9):1526–1538, Sep 2013. doi: 10.1109/TVCG.2013.65 2

[17] K. Misue, P. Eades, W. Lai, and K. Sugiyama. Layout adjustment and the mental map. *Journal of Visual Languages & Computing*, 6(2):183–210, June 1995. doi: 10.1006/jvlc.1995.1010 4

[18] V. Molchanov and L. Linsen. Smooth map deformation using integral images. *Journal of WSCG*, 28(1–2):18–26, July 2020. doi: 10.24132/JWSCG.2020.28.3 2

[19] F. V. Paulovich and R. Minghim. HiPP: A novel hierarchical point placement strategy and its application to the exploration of document collections. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1229–1236, November 2008. doi: 10.1109/TVCG.2008.138 2

[20] R. G. Raidou, M. E. Gröller, and M. Eisemann. Relaxing dense scatter plots with pixel-based mappings. *IEEE Transactions on Visualization and Computer Graphics*, 25(6):2205–2216, June 2019. doi: 10.1109/TVCG.2019.2903956 2

[21] H. Rave, V. Molchanov, and L. Linsen. De-cluttering scatterplots with integral images. *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–13, 2024. doi: 10.1109/TVCG.2024.3381453 1, 2, 3, 4

[22] B. D. Ripley. The second-order analysis of stationary point processes. *Journal of Applied Probability*, 13(2):255—-266, 1976. doi: 10.2307/3212829 3

[23] Tat-Jun Chin, Hanlin Goh, and Ngan-Meng Tan. Exact integral images at generic angles for 2D barcode detection. In *2008 19th International Conference on Pattern Recognition*, pp. 1–4, Dec 2008. doi: 10.1109/ICPR.2008.4761169 2

[24] J. Venna and S. Kaski. Neighborhood preservation in nonlinear projection methods: An experimental study. In G. Dorffner, H. Bischof, and K. Hornik, eds., *Artificial Neural Networks—ICANN 2001*, pp. 485–491. Springer, Berlin, 2001. doi: 10.1007/3-540-44668-0_68 4

[25] P. Viola and M. Jones. Robust real-time object detection. *International Journal of Computer Vision*, 57(2):137–154, 2002. doi: 10.1023/B:VISI.0000013087.49260.fb 2

[26] M. O. Ward. A taxonomy of glyph placement strategies for multidimensional data visualization. *Information Visualization*, 1(3-4):194–210, 2002. doi: 10.1057/PALGRAVE.IVS.9500025 2

[27] A. Woodruff, J. Landay, and M. Stonebraker. Constant density visualizations of non-uniform distributions of data. In *Proceedings of the 11th Annual ACM Symposium on User Interface Software and Technology*, UIST '98, pp. 19—-28. Association for Computing Machinery, New York, NY, USA, 1998. doi: 10.1145/288392.288397 1

[28] J. Yuan, S. Xiang, J. Xia, L. Yu, and S. Liu. Evaluation of sampling methods for scatterplots. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):1720–1730, 2021. doi: 10.1109/TVCG.2020.3030432 2