Improving 2D scatterplots effectiveness through sampling, displacement, and user perception

Enrico Bertini, Giuseppe Santucci
Dipartimento di Informatica e Sistemistica - Università di Roma "La Sapienza"
Via Salaria, 113 - 00198 Roma, Italy - {bertini, santucci}@dis.uniroma1.it

Abstract

In this paper we present a novel, hybrid, and automatic strategy whose goal is to reduce the 2D scatter plot cluttering. The presented technique relies on a combination of non uniform sampling and pixel displacement and it is driven by perceptual results coming from a suitable user study. The same results are used to define precise quality metrics that allow for validating our approach.

1 Introduction

Visualizing a data set containing large amounts of data likely produces a cluttered image. The user suffers from a strong sense of mess that rises from both the intrinsic limits of visual devices and adopted techniques. As the number of items increases, almost any kind of visual technique fails to convey detailed information; a lot of graphical elements overlap and many pixels become over plotted, losing useful pieces of information.

This paper focuses on 2D scatter plots extending and merging the results presented in [3, 5, 4]. In particular, we analyze data density that is one of the main clues the user can grasp from such a kind of visualization and, in order to reduce clutter, we sample the data in a way that preserves, as much as possible, density differences.

We address the cluttering problem in the following way:

- we defined a formal framework to measure the amount of degradation resulting from a given visualization, characterizing, e.g., the notion of collision and density, both in a virtual space and a real space. Moreover, such a formal framework allows for forecasting the number of collisions and the screen occupation. In this paper we recall just the main definitions, details about the matter are in [4].
- using the above framework we defined some quality metrics able to characterize the image degradation and to drive corrective actions.
- we set up a set of strategies to reduce the image cluttering, using different kinds of sampling, i.e., uniform sampling and non uniform sampling. Uniform sampling relies on the idea of to randomly sample the data set till a quality metric(s) reaches a predefined value (details about the matter are in [3]). Non uniform sampling, discussed in [4], considers the difference in densities that exists in the dataset and applies different sampling rates to different image areas, trying to maximize the number of density differences available on the screen.
- we analyzed the way in which users perceive density differences through a user study [5]. This allowed for fine tuning our metrics considering perceptual density differences instead of numerical differences. A first application of of this results is presented in [5] in which the uniform sampling strategy is driven by the new metric.
- we revised the non uniform sampling technique introducing two innovations: (a) we modified the overall algorithm strategy incorporating the user study results and we integrated in it a novel local pixel displacing strategy. These two last improvements represents the main contribution of this paper and are discussed in detail in Section 5.

Summarizing, the contribution of the paper is twofold:

1. it presents a novel technique that uses at the same time pixel displacements and non uniform sampling;
2. it exploits perceptual quality metrics for (a) estimating the image degradation and (b) driving automatic enhancing techniques.

The paper is structured as follows: Section 2 analyzes related works, Section 3 recalls the user test results, Section 4 introduces several quality metrics, Section 5 describes our sampling/displacing technique, Section 6 discuss the obtained results, and, finally, Section 7 presents some conclusions and open problems.

Proceedings of the Ninth International Conference on Information Visualisation (IV'05)
1550-6037/05 $20.00 © 2005 IEEE

2 Related work

Our methods use quality metrics, sampling, and displacement, as a way to produce more accurate visualization and to reduce clutter. In the following we first report on related proposals on metrics for Information Visualization, then we relate our approach to clutter reduction to some other existing ones.

2.1 Metrics and perceptual issues

The fact that Information Visualization needs metrics to provide precise indications on how effectively a visualization presents data is well known. As expressed in [14] Information Visualization needs methods to measure the "*goodness*" of a given visualization and a definitive and strong set of methodologies/tools is still lacking.

First attempts towards this direction come from Tufte that in [17] proposes an interesting set of measures to estimate the "graphical integrity" of static (i.e., paper based) representations. Brath, in [15], starting from Tufte's proposal, defines new metrics for static digital 3D images. He proposes metrics such as *data density* (number of data points/number of pixels) that recall Tufte's approach. He provides metrics aiming at measuring the visual image complexity like the *occlusion percentage*, that is the number of occluded elements in the visual space (having interesting connections with our metrics), or the *number of identifiable points*, that is the number of visible data points whose position is identifiable in relation to every other visible data point. While the main goal of the above metrics is to estimate a general visualization goodness, or to compare different visual systems, we mainly aim to assess the accuracy of a specific visualization, dealing with pixels and data points, in order to measure how accurately a visualization represents some data characteristics we are interested in (e.g., data density).

Measuring, in this context, means measuring the perception of visual features, therefore perceptual issues must be taken into account. Many studies have been conducted in the past to increase the effectiveness of visual systems and to avoid degradation. Results coming from color theory have been applied in practice in the context of data visualization to select color scales that appropriately reflect the properties of underlying data [12][2]. Preattentive features (visual features detected by the human eye without cognitive workload) have been exploited in the visualization of multivariate data to allow the users to efficiently detect visual patterns. Healey, in various proposals, inspected the effectiveness of preattentive features in deep and applied the results to build visually effective and efficient visualizations [9]. In this paper, we exploit the results of a perceptual study we conducted and presented in [5], whose aim was to understand how users perceive density differences in point based 2d scatterplots. With these results, we are able to detect the threshold values beyond which density differences are perceived.

2.2 Dealing with clutter

The problem of reducing visual clutter to produce more effective visual representations has been directly and indirectly addressed by a variety of proposals. Some of them deal with the problem of reducing the overall clutter of the visualization, especially when the screen displays a large number of items, while others try to resolve clutter locally.

Clustering has been used as a way to aggregate visual items to reduce the overall density, thus reducing clutter. Hierarchical parallel coordinates is one example of such a method [8].

Jittering is used in commercial systems like Spotfire [1]: the overlapping items are displaced around their original position so that they become visible [13]. Trutschl et al. propose a smart jittering technique [16]: jittering is applied in a way that items that are similar in the n-dimensional data space are close with one another when moved from their original position. We also use a kind of jittering in our technique. But, while the typical jittering displaces items around their original position randomly and applies it to the whole image, we use a selective jittering that runs only on specific areas and tries to move the items as less as possible from their original position. PixelMap [10][11] uses the same idea of displacing items around their original position together with a controlled distortion. It is used in geographical applications where each pixel represents the measure of some variable in a given location. The distortion introduced by the displacement is balanced by the distortion of the map (e.g., the boundaries between states) so that topological relationships are maintained.

Constant density visualization [18][19] is a distortion approach which is more oriented towards the representation of distorted overviews to deal with clutter. It presents more details within less dense areas and less details within denser ones, allowing the screen space to be optimally utilized and to reduce clutter. The drawbacks of this approach are that (a) it requires the user to interact with the system, (b) the overall trend of data is generally lost, and (c) some distortions are introduced.

Sampling is used in [6][7] as a way to reduce clutter. Since sampling reduces the number of displayed elements, the overall visual density decreases and the visualization becomes more intelligible. Uniform sampling has the interesting benefit that data features like distribution and correlation are preserved, allowing *"to see the overall trends*

in the visualization but at a reduced density". However, this idea is not free of drawbacks. In particular, choosing the right amount of sampling is a challenging task and a straightforward solution does not exist, leaving to the user the non easy task of interactively selecting a sample ratio. Moreover, when data present both very high and very low density areas, two problems can arise:

- if sampling is too *strong* the areas in which density is very low become completely empty;
- if sampling is too *weak* the areas with highest densities still look all the same (i.e., completely saturated) and, consequently, density differences cannot be perceived.

In summary, our approach differs from the discussed metrics and clutter reduction techniques for three main aspects: it provides a sound model for defining, both in a virtual and physical space, several metrics specifically intended for digital images; it provides some *quantitative* information about an image quality; it exploits such results to automatically drive sampling algorithms preserving, as much as possible, specific visual characteristics.

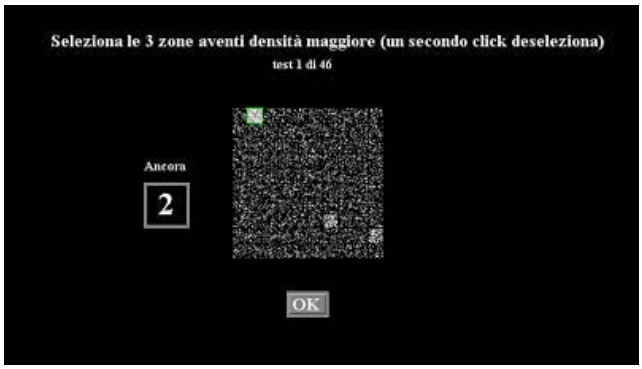


Figure 1. The user study main screen.

Basis\Delta	D1	RP1	D2	RP2	D3	RP3	D4	RP4	D5	RP5	DMIN
10	55	62	65	77	75	82	85	92	95	97	**65**
20	35	41	40	64	45	70	50	77	55	87	**45**
30	30	62	35	56	40	74	45	95	40	97	**40**
40	26	67	30	77	34	85	38	90	42	100	**30**
50	20	59	22.5	79	25	77	27.5	92	30	95	**22.5**
60	22	72	25	92	28	100	31	97	34	100	**22**
70	12	64	13.5	87	15	73	16.5	77	18	90	**15**
80	10	70	11.5	87	13	95	14.5	97	16	97	**10**
90	6	77	7	92	8	100	9	97	10	100	**6**

Figure 2. The user study results (all values are percentages).

3 The User Study

We performed a perceptual user study in order to answer a precise question: what is the minimum difference in active pixels between two screen areas that allows a user to perceive a density difference? In this paper we omit the experiment details and we just recall the main experiment guidelines and results; the interest reader can found in [5] a full experiment description.

The main idea underlying the experiment was to present the involved people with a uniform density screen (basis) containing three more dense zones and to ask the users to recognize them. We repeated the test, for each subject, several times increasing both the uniform background density and the density difference (delta) between the background and the three denser zones. Figure 1 shows a generic experiment step, in which the user marked one of the three denser areas.

The results are collected in the table shown in Figure 2. The table shows per each row a different basis (10, ..., 90) and the five different increment steps (δ) adopted along the test (D1, ..., D5); for each increment the table shows the corresponding recognition percentage (RP1, ..., RP5) as well. As an example, the first row tells us that, while evaluating a basis of 10%, we asked the user to identify areas containing 55%, 65%, 75%, 85%, and 95% extra pixels and that the recognition rate was 62%, 77%, 82%, 92%, and 97%, respectively. We performed a single factor ANOVA confirming the significance of our figures. The last column shows, for each basis, the minimum increment we have to choose to guarantee that 70 out of 100 end users will perceive the density difference. Using such a column we can interpolate a function $minimum\delta(A)$ returning the minimum density increment an area A' must show to be perceived as denser than A (by 70 out of 100 users).

The experiment results have been used to improve the accuracy of our algorithms, as described in Sections 4 and 5.

4 Models and metrics for density differences

We consider a 2D space in which we plot items by associating a pixel to each data element and the pixel position is computed mapping two data attributes on the spatial coordinates. In [4] we derived a complete framework to estimates the amount of colliding points and, as a consequence, the amount of free available space; here we just recall the main definitions useful for the following discussion. Moreover, we introduce a substantial modification to our quality metrics to take into account the user perception.

4.1 Definitions

We assume the image is displayed on a rectangular area (measured in inches) and that small squares of area A divide

the space in $m \times r$ *sample areas* (SA) where density is measured. Given a particular monitor, the resolution and size affect the values used in calculations. In the following we assume that we are using a monitor of 1280x1024 pixels and size of 13"x10.5". Using these figures we have 1,310,720 available pixels and if we choose SA of side $l = 0,08$ inch, the area is covered by 20.480 (160×128) sample areas whose dimension in pixels is 8×8.

For each $SA_{i,j}$, where $1 \leq i \leq m$ and $1 \leq j \leq r$, we calculate two different densities : *real data density* (data density in the following) and *represented density*.

Data density is defined as $d_{i,j} = \frac{n_{i,j}}{A}$ where $n_{i,j}$ is the number of data points that fall into sample area $A_{i,j}$. For a given visualization, the set of data densities is finite and discrete. In fact, if we plot n data elements, each $SA_{i,j}$ assumes a value $d_{i,j}$ within the set $0, \frac{1}{A}, \frac{2}{A}, \ldots, \frac{n}{A}$. In general, for any given visualization, a subset of these values will be really assumed by the sample areas. For each distinct value we can count the number of sample areas computing the data density distribution. For example, if we plot 100 data points onto an area of 10 sample areas, we could have the following distribution: 3 sample areas with 20 data points, 2 sample areas with 15 data points, 2 sample areas with 5 data points.

Represented density is defined as $rd_{i,j} = \frac{p_{i,j}}{A}$ where $p_{i,j}$ is the number of distinct active pixels that fall into $SA_{i,j}$. The number of different values that a represented density can assume depends on the size of sample areas. If we adopt sample areas of 8×8 pixels the number of different not null represented densities is 64.

Because of collisions the number of active pixels on a sample area $SA_{i,j}$ will be smaller than the plotted point so $RD_{i,j} \leq D_{i,j}$.

4.2 Quality metrics

In the following we provide a quality metric that, focusing on the distorted area, provide an indication on how many density differences are still visible in the displayed image.

The complete list of the involved parameters is the following:

- the overall number of points being plotted, n;
- the display area size, in terms of number of pixels, x_pixels, y_pixels;
- the sample areas size in terms of number of pixels, l_pixels (we are considering squared sample areas);
- the number of collisions k per sample area (SA);
- the data density and the represented density.

In order to introduce our metric we need some preliminary measures and definitions. In particular, because of our metrics focuses on distorted areas, we introduce a threshold value Δ that allows for distinguishing acceptable crowded SAs from non acceptable ones. To fix the idea, we can state that we cannot bear SAs showing more than 32% of collisions w.r.t. $l\_pixels \times l\_pixels$. Obviously, the lower this value the better the image and Δ is a parameter that allows for fine-tuning our algorithms.

Using Δ we can define the following metric:

$$\text{BSAr(Bad SA ratio)}=\frac{\#\ of\ SA\ showing\ k>\Delta}{\#\ of\ SA}$$

that gives the measure of the screen percentage affected by a non acceptable distortion.

Now we can concentrate on relative densities, measuring the density differences that are preserved in the distorted portion of the image through the metric PDDr (Preserved Data Densities ratio). This metric is calculated comparing couples of sample areas and checking whether their relative data density (D) is preserved when considering their represented density (RD).

Introducing the $Diff(x, y)$ function defined as:

$$Diff(x,y) = \begin{cases} 1 & \text{if } x > y \\ 0 & \text{if } x = y \\ -1 & \text{if } x < y \end{cases}$$

we define the $match(i, j, k, l)$ function that returns true iff $Diff(D_{i,j}, D_{k,l}) = Diff(RD_{i,j}, RD_{k,l})$.

In order to produce a measure, we apply an algorithm that iteratively considers all the possible couples of Distorted SAs (DSA), comparing their D and RD through the $Diff$ function and counting the number of times it finds a non matching pair.

Moreover, in order to take into account the relevance of a comparison between two sample areas, we weight each comparison using the number of points falling in the two sample areas.

In pseudo-code, the algorithm is the following:

```
function PDDr(){
 couples=0;
 sum=0;
 foreach distinct pair(DSA[i][j], DSA[k][l]){
   couples = couples + pt(DSA[i][j]) + pt(DSA[k][l]);
   if ( match(i, j, k, l) )
      sum = sum + pt(DSA[i][j])+ pt(DSA[k][l]);
 }
 return (sum / couples);
}
```

where $pt(SA_{i,j})$ is a function returning the number of data points falling in a SA.

In the end the variable *sum* contains the number of weighted matchings couples encountered during the iterations; dividing it by the weighted total number of possible

distinct distorted couples we obtain the weighted percentage of matching sample areas ranging between 0 and 1 (the higer the better).

This last metric provides a distortion evaluation, counting the densities differences still visible in the crowded area and weighting such differences through the involved points.

The main drawback of this metric is that it uses numerical differences between sample areas to decide wether a data density difference is well represented by the corresponding represented densities. As an example, a sample area containing 55 active pixels is considered denser than another one containing 54 active pixels while both of them look the same to the end user.

The experiment results have been used to improve our metrics and algorithms, introducing the $PDiff(x, y)$ (Perceptual Diff) function as a modification of the above introduced $Diff(x, y)$ function:

$$PDiff(x,y) = \begin{cases} 1 & \text{if } x \geq y + y \times minimum\delta(y) \\ -1 & \text{if } y \geq x + x \times minimum\delta(x) \\ 0 & \text{otherwise} \end{cases}$$

Using the PDiff function within the match function modifies in a substantial the above metric, obtaining the PPDDr (*Perceptually* Preserved Data Densities ratio) metric. In this way the quality metric deals with user perceptible vs numeric density differences. It is worth noting that the new $PDiff(x, y)$ function returns a lower number of matching than the $Diff(x, y)$ function, allowing the optimization algorithm to focus only on what really matters: the user point of view.

In order to better understand the difference between the two approaches, we apply the two metrics against the example in Figure 3 that shows about 160,000 mail parcels plotted on the X-Y plane according to their weight (X axis) and volume (Y axis). It is worth noting that, even if the occupation of the screen is very little, the area close to the origin is very crowded (usually parcels are very light and little), so a great number of collisions is present in that area.

Using the pure numeric metric, PDDr with the 32% collision delta, we obtain the reasonable vale of 0.71, meaning that in the distorted area about 71% of the data points are presented correctly in the image (i.e., their relative density is preserved in the final image). If we consider, instead, the PPDDr metric we obtain a worse (but more realistic) value, 0.57. That implies that the old metric counted a great number of "fake" density differences (numerical differences) not perceivable by the users.

5 Sampling and displacement to reach target densities

The basic problem we want to address here is to find a way to represent, in the limited visualization space, as many density differences as we can, that is, trying to provide a visualization that is as representative as possible of the real underlying densities. In general, for visualizations where specific interventions are not employed, this correspondence is not accurate: when data are visualized on a crowded scatterplot, high data densities are mapped to few represented densities, the ones in which almost all pixels are active, and a large number of high data densities are "squeezed" to few and very close represented densities; thus, some existing density differences cannot be perceived.

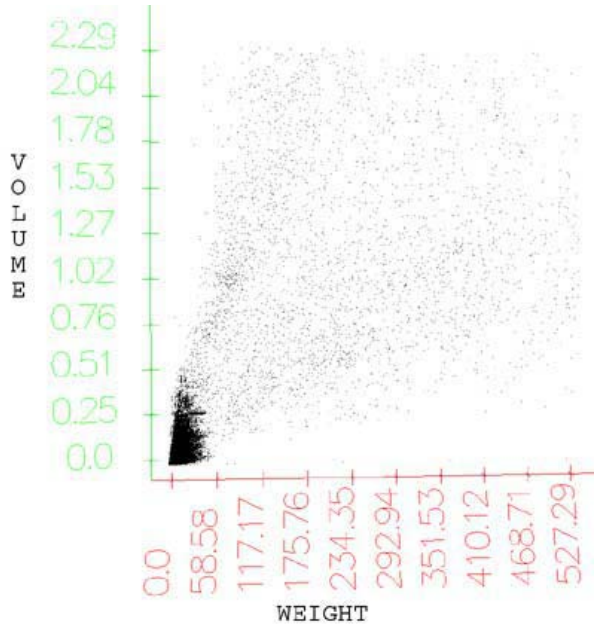


Figure 3. The scatter plot represents a dataset of mail parcels. The parcel's weight is mapped to the X-axis and the parcel's volume to the Y-axis.

Any given visualization is a particular mapping between the set of data densities and the set of available represented densities, therefore the problem can be translated into the one of finding an optimal mapping between data densities and represented densities, that is, associating each data density to one of the 64 (under the hypothesis of 8×8 sample areas) available represented densities. Modifying this mapping, we can potentially find some more accurate representations. To this aim, we must: (1) find a method to decide which data densities are mapped to which represented densities; (2) then we need a way to perform these mappings in practice.

As for the first point, we devised an algorithm that splits the range of existing data densities into 64 intervals which will be assigned to the 64 available represented densities. It calculates the average number of sample areas with respect

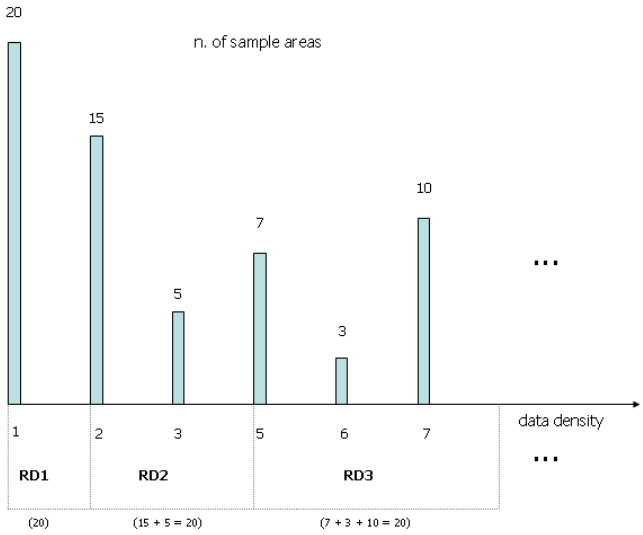


Figure 4. The split algorithm. The data density axe is split into 64 values in a way that each contains the same number of sample areas.

to the available represented densities ($K = \frac{n.sampleareas}{64}$), then, using this value, it builds the intervals. Starting from data density $D = 1$ ($D = 0$ is not taken into account in calculations because no interventions are applied in empty sample areas) it adds densities to the current interval until the sum of the included sample areas is equal to the average. When the value K is reached, a new interval is built. In the example in Figure 4, $D = (1)$ is assigned to $RD = 1$ because it already spans 20 sample areas; $D = (2, 3)$ are assigned to $RD = 2$ because they sum up to 20; $D = (7, 3, 10)$ are assigned to $RD = 3$ because they sum up to 20; and so on. The resulting effect is that we have intervals of different sizes: large intervals containing many data densities that span few sample areas; short intervals containing few data densities that span many sample areas. This implies that the densities that cover many sample areas are sampled with finer details, because it is more likely they are singularly mapped to single represented densities, while the opposite happens for the densities that cover few sample areas. The rationale behind this is that the algorithm tries to represent more accurately the densities that span a large portion of the screen while it accepts some distortion for few concentrated areas.

Once the mapping is computed we have, for each data density, a target density to reach. Thus, we need a way to turn on the number of pixels that produces the desired represented density. There are three possible cases and three associated interventions:

Represented density is equal to target density. This is the simplest case, the current represented density is already *equal* to the one we want to reach, so we just have to leave things as they are.

Represented density is greater than target density. This is the case when the number of pixels turned on by the data points falling in the current sample area is *higher* than the target density we want to reach. In order to change the represented density we sample the data until we reach the target represented density.

Represented density is lower than target density. This is the case when the number of pixels turned on by the data points falling in the current sample area is *lower* than the target density we want to reach. It is worth to note that this case can happen only because of data points' collision. In our model, data density is always higher than any target density, therefore if the current mapping provides an insufficient number of active pixels, these must necessarily be because of overlapping. In order to reach the target density, we use pixel displacement so that some overlapping items become visible and represented density can be increased. In order to minimize the distortion introduced by displacement, the pixels are moved as close as possible to their original position. In any case, since the displacement takes place locally, within single sample areas, the entity of distortion is minimal and cannot have macroscopic negative effects.

When the right mapping has been performed and target densities are obtained we have, in principle, the best possible mapping. Looking at the image through the lens of our $PDDr$ metric (see Section 4.2) this is the best result we can achieve. But, as pointed out in Section 3, this trail of thoughts does not take into account the fact that differences of one single pixel cannot be perceived. This is why a third stage is needed.

Using the results given in Section 3, we restrict the values of represented densities to the ones that can perceived as different. To this aim, we re-sample the dataset in a way that only the perceivable represented densities are presented on the screen, that is, the ones obtained in the user study. Starting from the represented density 64 downward to 1, they are sampled to let them reach their next available lower value: (1) to 1; $(2, 3)$ to 2; $(4, 5, 6)$ to 4; etc. The complete set of mappings is reported in Table 1.

Roughly speaking, we can think of the whole process as follows. We have at disposal p different represented densities that are matched against k real data densities where, likely, $k >> p$; that implies that each represented density is in charge to represent several different data densities, hiding differences to the user. The strategy consists in changing, with sampling and displacement, the original data densities, altering their assignment to the p available represented den-

Represented Density	Perceptual Density
1	1
2, 3	2
4, 5, 6	4
7,8,9,10	7
11,12,13,14,15,16	11
17,18,19,20,21,22,23	17
24,25,26,27,28,29,30,31	24
32,33,34,35,36,37,38	32
39,40,41,42,43,44,45,46	39
47,48,49,50,51,52	47
53,54,55,56,57	53
58,59,60	58
61,62,63	61
64	64

Table 1. Represented densities mapped to perceptual densities. In order to visualize only visible density differences, the represented densities are re-sampled according to the values provided in the table.

sities to maximize the number of correctly represented data density differences. Then, since the problem of perceptual differences is recognized, an additional step is introduced in which the visualization is re-sampled to obtain only perceivable represented differences.

6 Discussion

In order to assess the validity of our method we compare the new technique with the old one, using a real dataset that contains 160,000 mail parcels data. Figure 5 shows: (a) the original visualization, (b) the one obtained with the old method (i.e., non-uniform sampling), (c) the one obtained using the new method (i.e., with sampling, displacement, and perceptual issues).

It is quite evident that both Figure 5 (b) and (c) allows for grasping more details in the most crowded zone, leaving the less dense areas quite untouched. This is a general characteristic of the non uniform sampling. Moreover, looking at Figure 5 (c) we can note two additional improvements:

1. the density differences are more evident. This is due to the fact that not all the represented densities are available: according to Table 1, only 14 out of 64 represented densities are available on the screen and this, according to the user study, increases the density differences perception;

2. some faint areas are more evident. As an example, in the left lower part of the image is quite evident a new cluster. This is the effect of the displacement activity that rescues points in areas that present a number of collisions higher than the average.

The visual impressions are confirmed by our metrics: the original values of PDDr and PPDDr (0.71 and 0.57) rises in image (b) to 0.79 and 0.65 reaching, in image (c), the values 0.83 and 0.76. We can claim that our technique increased the percentage of data points that are correctly perceived by the end user (in terms of relative density) as much as 34% (i.e., 0.766/0.57).

7 Conclusion And Future Work

In this paper we presented a novel automatic strategy for enhancing 2D scatter plot quality, preserving in the final image as many density differences as possible. To this aim, the strategy incorporates three novel techniques:

1. it uses at the same time sampling and pixel displacement;

2. it is driven by perceptual issues gathered from a suitable user study;

3. it incorporates a quality metric useful for both driving the algorithm and validating the results.

Several open issues rise from this work. In particular, several choices deserve more attention: it is our intention to analyze the influence of increasing/decreasing of sampling area dimension, in term of image quality and computational aspects. Moreover, some statistical analysis of the data distribution could provide automatic means for determining the optimal sample area dimension.

Finally, we are we are currently extending the presented approach to other well known Infovis techniques, i.e., parallel coordinates.

8 Acknowledgements

This work has been supported by the DELOS Network of Excellence on Digital Libraries and MAIS "Multichannel Adaptive Information Systems" projects. We would like to thank Gabriele Gorini for his invaluable help in implementing the software prototype and tuning the sampling algorithms.

References

[1] Christopher Ahlberg and Erik Wistrand. Ivee: an information visualization and exploration environment. In *Proc. of IEEE Symposium on Information Visualization*, page 66. IEEE Computer Society, 1995.

[2] L. D. Bergman, B. E. Rogowitz, and L. A. Treinish. A rule-based tool for assisting colormap selection. In *Proceedings of the 6th conference on Visualization '95*, page 118. IEEE Computer Society, 1995.

[3] E. Bertini and G.Santucci. Quality metrics for 2d scatterplot graphics: automatically reducing visual clutter. In *Proceedings of 4th International Symposium on SmartGraphics*, May 2004.

[4] Enrico Bertini and Giuseppe Santucci. By chance is not enough: preserving relative density through non uniform sampling. In *Proc. of IEEE Information Visualization Conference*, July 2004.

[5] Enrico Bertini and Giuseppe Santucci. Is it darker? improving density representation in 2d scatter plots through a user study. In *Proc. of SPIE Conference On Visualization and Data Analysis*, January 2005.

[6] G. Ellis and A. Dix. Density control through random sampling: an architectural perspective. In *Proceedings of Conference on Information Visualisation*, pages 82–90, July 2002.

[7] Geoff Ellis and Alan Dix. by chance: enhancing interaction with large data sets through statistical sampling. In *Proc. of ACM Working Conference on Advanced Visual Interfaces*, pages 167–176. ACM Press, may 2002.

[8] Ying-Huey Fua, Matthew O. Ward, and Elke A. Rundensteiner. Hierarchical parallel coordinates for exploration of large datasets. In *Proc. of Visualization'99*, pages 43–50. IEEE Computer Society Press, 1999.

[9] Christopher G. Healey and James T. Enns. Large datasets at a glance: Combining textures and colors in scientific visualization. *IEEE Transactions on Visualization and Computer Graphics*, 5(2):145–167, 1999.

[10] Daniel A. Keim, Christian Panse, Joern Schneidewind, and Mike Sips. Geo-spatial data viewer: From familiar land-covering to arbitrary distorted geo-spatial quadtree maps. In *WSCG 2004, The 12th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, February.

[11] Daniel A. Keim, Christian Panse, Mike Sips, and Stephen C. North. Visual data mining in large geospatial point sets. *IEEE Computer Graphics and Applications*, 24(5):36–44, 2004.

[12] Haim Levkowitz and Gabor T. Herman. Color scales for image data. *IEEE Computer Graphics and Applications*, 12(1):72–80, 1992.

[13] Jeremy Manson. Occlusion in two-dimensional displays: Visualization of meta-data. Technical report, University of Maryland, College Park, 1999.

[14] Nancy Miller, Beth Hetzler, Grant Nakamura, and Paul Whitney. The need for metrics in visual information analysis. In *Proceedings of the 1997 workshop on New paradigms in information visualization and manipulation*, pages 24–28. ACM Press, 1997.

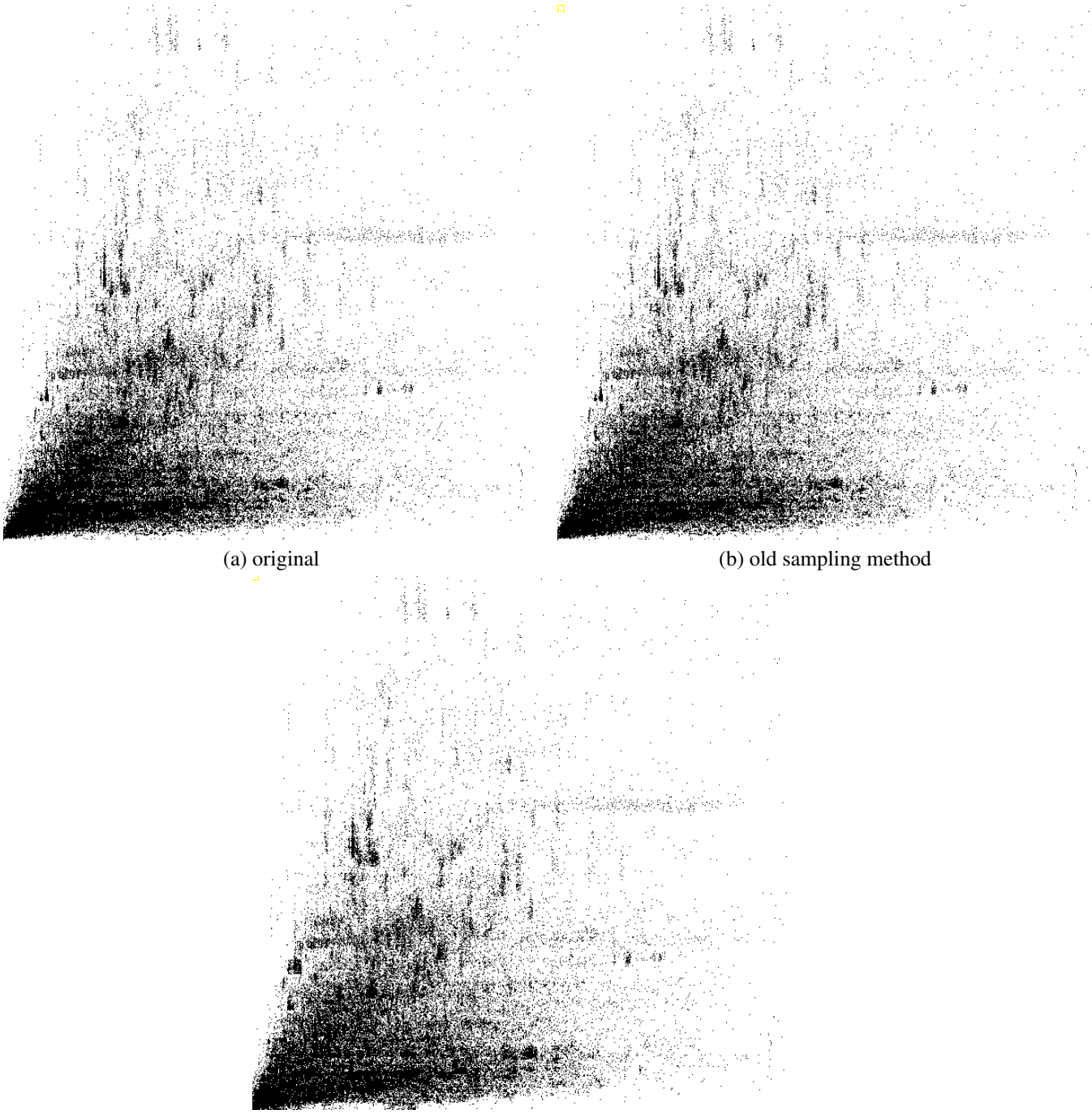
[15] Brath Richard. Concept demonstration: Metrics for effective information visualization. In *Proceedings For IEEE Symposium On Information Visualization*, pages 108–111. IEEE Service Center, Phoenix, AZ, 1997.

[16] Marjan Trutschl, Georges Grinstein, and Urska Cvek. Intelligently resolving point occlusion. In *Proceedings of the IEEE Symposium on Information Vizualization 2003*, page 17. IEEE Computer Society, 2003.

[17] Edward R. Tufte. *The visual display of quantitative information*. Graphics Press, 1986.

[18] Allison Woodruff, James Landay, and Michael Stonebraker. Constant density visualizations of non-uniform distributions of data. In *Proceedings of the 11th annual ACM symposium on User interface software and technology*, pages 19–28. ACM Press, 1998.

[19] Allison Woodruff, James Landay, and Michael Stonebraker. Vida: (visual information density adjuster). In *CHI '99 extended abstracts on Human factors in computing systems*, pages 19–20. ACM Press, 1999.



(a) original

(b) old sampling method

(c) new method

Figure 5. Comparison of sampling methods visualizing the mail parcels dataset: (a) the original visualization; (b) the visualization obtained with our old sampling method (i.e., perceptual issues neglected); (c) the visualization obtained with our new method taking into account perceptual issues.