Give chance a chance: modeling density to enhance scatter plot quality through random data sampling

Enrico Bertini Giuseppe Santucci

Corresponding author: Enrico Bertini

Address: Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza" Via Salaria, 113, 00198 Rome, Italy

Email: bertini@dis.uniroma1.it

Phone: +39-06-49918339

Fax: +39-06-85300849

Short running title: Overplotting Reduction with Random Sampling

Acknowledgement: Work supported by the MIUR-FIRB project "MAIS" (Multichannel adaptive Information Systems, http://www.mais-project.it/)

Abstract

The problem of visualizing huge amounts of data is well known in Information Visualization. Dealing with a large number of items forces almost any kind of Infovis technique to reveal its limits in terms of expressivity and scalability. In this paper we focus on 2D scatter plots, proposing a "feature preservation" approach, based on the idea of modeling the visualization in a virtual space in order to analyze its features (e.g, absolute density, relative density, etc.). In this way we provide a formal framework to measure the visual overlapping, obtaining precise quality metrics about the visualization degradation and devising automatic sampling strategies able to improve the overall image quality. Metrics and algorithms have been improved through suitable user studies.

Keywords: overplotting, sampling, quality metrics, numerosity

1 Introduction

Visualizing a large data set likely produces a noisy image: many pixels become overplotted, losing useful data insights. In this paper we attack this problem focusing on 2D scatter plots, a very useful and widely adopted visualization technique. In particular, we analyze data density, one of the main clues the user can grasp from such a kind of visualization and, in order to reduce overplotting, we sample the data preserving, as much as possible, density related aspects.

We address the overplotting problem in two steps: first we provide a formal framework to measure the degradation affecting a given visualization, then, upon these measures, we sample the data improving the image quality.

To measure the image degradation we define a formal model that estimates the amount of overlapping elements in a given area and, consequently, the remaining free space. These pieces of information give an objective indication of what is eventually visualized on the physical device and we can compute the quality of the displayed image.

To reduce overplotting we employ different sampling techniques taking into account *how much* and *where* to sample in order to preserve interesting features. In fact, the formal model we discuss in the paper gives precise indications on the right amount of data sampling needed to produce a representation preserving the most important image characteristics. We use two different sampling techniques, *best uniform sampling* and *non uniform sampling*. Best uniform sampling tries to present the user with as many density differences as possible, preserving the magnitude of such differences; non uniform sampling increases the number of density differences available on the screen altering their magnitude.

To improve our sampling algorithms we analyze the way users perceive density differences through a user study that investigates the correspondence between numerical density differences and users' perception of density differences. This allows for fine tuning our metrics, considering perceptual density differ-

ences instead of numerical ones.

Moreover, as discussed in Section 8, using the tool we developed for testing our metrics and algorithms, we discovered that the availability of different sampling techniques, revealing different data insights, facilitates interactive explorative analyses.

Summarizing, the contribution of the paper is twofold:

1. it presents a novel model that allows for defining and measuring data density both in terms of a virtual space and of a physical space;

2. it defines user validated quality metrics that allow for (a) estimating the image degradation and (b) driving novel automatic sampling techniques.

1.1 Paper organization

The paper is structured as follows: Section 2 analyzes related works, Section 3 describes the model we use to characterize overplotting and density, Section 4 describes the user study, Section 5 introduces several quality metrics, Section 6 describes our sampling techniques, Section 7 reports implementation issues, Section 8 outlines possible extensions of our approach, and, finally, Section 9 presents some conclusions.

2 Related work

The main objective of our investigation is to provide models and techniques to deal with overplotting, exploiting quality metrics and sampling strategies. In the following we first report on related perceptual studies, then we discuss metric proposals for Infovis, and, finally, we provide on overview on techniques to cope with overplotting.

2.1 Density Perception

Central to our purposes is the visual phenomenon called "numerosity" which explains how people perceive relative densities. When an observer has to judge how many items are on a scene, and there are too many objects and/or time is too short to count, a person is still able to provide an approximate number which is called numerosity. Many studies have been conducted on the subject, presenting involved people with short sequences of random black dots and asking them to recognize the one containing more items.

Studies show that the spatial configuration of dots influences the perception of density so that, e.g., two figures with the same number of dots can be perceived as having a different number of items. The *occupancy model* partially explains these effects providing a model [30] in which each dot has an area of influence upon its neighborhood; the regions where the areas overlap and exceed a threshold value are considered as being filled with dots. The size of the area occupied by the dots seems to be irrelevant in the perception of numerosity [3] (but it has been tested on a limited range of values), whereas the dimension of dots seems to influence it in a inverse relation between size and numerosity [13]. Another relevant factor is the color. Beaudot et al. demonstrate in [31] that there is a "consistent bias in favor of blue-yellow stimuli which are perceived as significantly more dense than red-green and achromatic stimuli".

Various studies aimed at finding threshold values, i.e., given a certain number of base items, to discover how many additional ones are necessary to see a difference, and how this difference changes with the number of base items. Results show that numerosity increases with the number of items but not linearly [1], as often happens with psychophysics measures [27]. A peculiarity of numerosity judgement is that it does not follow the Weber's law [1] as one might

[1]The Difference Threshold (or "Just Noticeable Difference") is the minimum amount by which stimulus intensity must be changed in order to produce a noticeable variation in sensory experience. Weber's Law says that the size of the just notice-

expect: the ratio between the number of additional items necessary to see a difference and the number of items in the reference pattern is not constat; actually it decreases as the number of items increases. This means that the more we add items, the less is the percentage of new items needed to see a difference.

Without pretending to draw generalizable results for vision science or to extend the theory on numerosity, we conducted a series of experiments which resembles those classic studies, but targeted at our peculiar needs. The main aim of our experiments is to tweak our algorithms in order to take into account perceptual density differences in place of plain numeric ones. The description of our motivations, peculiar needs, and differences with existing studies are in Section 4, where the experiment is described in detail.

2.2 Metrics for Information Visualization

Providing quality metrics is a well known Infovis need: there is the necessity to objectively assess the quality of a visualization through formal measures [23].

First attempts come from Tufte that in [29] proposes a set of measures to estimate the "graphical integrity" of static (i.e., paper based) representations. Measures like the *lie factor*, that is the ratio between the size of an effect, as shown graphically, to its size in the data, or *data density* that takes into account the size of the graphic in relation to the number of displayed data, are examples of his attempt to systematically provide indications about the quality of the displayed image. Brath, in [24], 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*, 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

able difference is a constant proportion of the original stimulus value. (source: http://www.usd.edu/psyc301/WebersLaw.htm)

point. These metrics are interesting and are appropriate for characterizing digital images. However, as stated by the author, they are still immature. Another approach is that of using benchmarks to evaluate visualizations. Grinstein, in [14], proposes to run some benchmarks against predefined datasets and tasks to evaluate and compare different visual techniques.

While the main goals of the above method 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. Moreover, we provide precise quality metrics that can be directly exploited by recovery algorithms. In Section 5 we formally define such metrics and we show how to use them *in practice* to take quantitative decisions on corrective actions.

2.3 Dealing with overplotting

Overplotting is a common problem that affects many user interfaces: as a display hosts too many objects noisy visualizations may arise. It is hard to find general recovery strategies and to provide formal definitions of the problem because the spectrum of possible degradations is fairly broad and difficult to formalize.

The problem of how to deal with highly dense visualizations with overlapping items has been directly and indirectly addressed by a variety of proposals. Some of them deal with visualization overviews, especially when the screen displays a large number of items, while others try to resolve the problem locally, that is, focusing on interesting subregions.

A common solution is the use of density maps: visualizations in which data density is mapped to color intensity to communicate density variations. There exist some variants of the technique. A first attempt is in [5] where relational data are previously binned and aggregated to compute discrete regions of the screen where data density is depicted through volume rendering. The result is a 3D point-based visualization (as in the early attempts to project n-dimensional

data in three-dimensional projection provided in PRIM9 [9]) with variable intensity and color. An almost identical approach is used by Yang in [35], where the method is used in conjunction with clustering and interactive picking and brushing to allow effective and efficient exploration. Another solution is the *Information Mural* [17], a general purpose technique that maps data overplotting to pixel's intensity or color to communicate data density. The method permits to cope with data overplotting in a variety of visualizations like time series, scatter plots, maps, etc. A similar approach can also be used when the primary visual mark is not a dot but a line, as in parallel coordinates [4].

Clustering resolves the problem through abstraction [26][32]. It aggregates similar data items in groups, so that a single visual mark represents a series of objects rather than one single data item, thus visual density is reduced. Hierarchical parallel coordinates uses hierarchical clustering on parallel coordinates [16]: each cluster is represented by a single poly-line with a surrounding halo that depicts its size [10] [11].

Jittering is used in commercial systems like Spotfire [2]; the overlapping items are displaced around their original position so that they become visible [22]. Trutschl et al. propose a smart jittering technique [28]: jittering is applied in a way that items that are similar in the n-dimensional data space stay closer when moved from their original position. PixelMap [18][19] 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.

Similarly, pure distortion techniques [12][21][20] can be effectively employed to resolve overplotting locally but they are almost useless in case of heavily crowded screens. Zoom can also be useful as a way to increase resolution for a limited area of the screen, but the overall context is often lost and complex interaction to navigate from one area to another my be required [15]. Constant density displays partially overcome the problem [33][34] presenting more details

within less dense areas, and less details within denser ones, allowing the screen space to be optimally utilized.

Sampling is used in [8][7] as a way to reduce density as well. 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"*.

2.4 Positioning our approach

Our approach differs from the discussed proposals for three main aspects:

1. it defines a sound model for defining, both in a virtual and physical space, several metrics specifically intended for digital images; such metrics allow for providing some *quantitative* information about an image quality;
2. it exploits such results to drive sampling algorithms preserving specific visual characteristics;
3. both metrics and algorithms are based on perceptual user studies.

As a consequence, our proposal presents some unique features, described in the following.

- Measuring lost features. Our formal framework and quality metrics allow for discovering, in a quantitative way, whether relevant data characteristics are preserved in the actual visualization; as an example, the metric PLDDr, described in the end of Section 5, measures the data density differences hidden from the user.
- Providing a detailed overview without altering the image size. Zooming, displacement, sampling, and density maps are commonly used to reduce overplotting. Displacing elements introduces some errors in the representation, while zooming causes to lose the overall image perception, unless

the zoom is applied to the whole image (in this case, the screen dimension represents the limit of this approach). Sampling and density maps may effectively improve the image readability without enlarging it; however, if the data present quite different density values these techniques are not very effective: faint zones may disappear while making the most dense areas readable and little density differences are hardly perceivable by the user. Our non uniform sampling technique, described in Section 6.2, addresses exactly this problem, providing a data overview preserving faint zones and showing details in denser areas, without altering the image size.

- Automatic image treatment. The availability of quality metrics allows for automatically ameliorate the image. As an example, it is possible to compute the optimum sampling ratio (w.r.t. a quality function) applying it to the image without user intervention (see Section 6.1). As another example, to optimize the screen usage, it is possible to compute the *minimum* image size that guarantees some quality threshold (see Section 5).

Finally, our approach can be used *together with* other techniques, producing interesting synergies. As an example, consider the gray scale density map presented in Information Mural [17]. As stated by the authors "Distinguishing fine variations or level of detail in a grey-scale is difficult for people"; applying our approach together with grey-scale could make Information Mural, or similar methods, more effective; similarly, applying together zoom and sampling reduces the amount of needed zoom, helping the user to not lose the context.

3 Modeling density

In this section we present a statistical framework characterizing the distortion produced by data overplotting, a typical problem that happens when a continuous space is represented in a discrete one.

The formal environment addresses two different objectives:

1. it allows for defining in a clear way all the image features considered in our approach, i.e., data density and represented density introduced at the end of this section;

2. it allows for foreseeing the number of active pixels and collisions resulting by applying a given amount of sampling on the actual data set. We will use such forecasts to drive the sampling algorithms described in Section 6.

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. As an example, Figure 1 shows about 160,000 mail parcels plotted on the X-Y plane according to their weight (X axis) and volume (Y axis). Note that, even if the occupation of the screen is very little, the area close to the origin is very crowded and presents a great number of collisions.

In the following we derive a function that estimates the amount of colliding points and, as a consequence, the amount of free available space. More formally, two data points are in collision when their projection is on the same physical pixel (likely for rounding issue); each time two points collide on the same pixel we count a collision, even if the pixel has been involved in other collisions: i.e., if n data points collapse on the same pixel we count $n - 1$ collisions.

In order to calculate the estimation function, we imagine to toss n data points in a completely random way (that is, the probability for each point to fall on a certain position is constant for any position) on a fixed area of p pixels. This assumption is reasonable if we conduct our analysis on *small* areas in which the real data distribution does not show large variations.

We consider the following parameters:

- n is the number of points we are plotting;

- p is the number of available pixels;

- k is the number of collisions;
- d is the number of free pixels.

The probability of having *exactly* k collisions plotting n points on an area of p pixels, $Pr(k, n, p)$, is given by the following formula:

$$\begin{cases} \frac{PERM[\binom{p}{n-k}\binom{n-k+k-1}{k}]}{p^n} & \text{if } n \leq p \text{ and } k \in [0, n-1] \\ & \text{or } n > p \text{ and } k \in [n-p, n-1] \\ \\ 0 & \text{if } k \geq n \\ & \text{or } n > p \text{ and } k \in [0, n-p-1] \end{cases}$$

The function returns zero if $k \geq n$ because it is impossible to have a number of collisions equal or greater than plotted points. Moreover, the probability is equal to zero if $n > p$ and $k \in [0, n-p-1]$: because we are plotting more points than available pixels, we must necessarily have at least $n - p - 1$ collisions. For example, if we have an area of 64 pixels and we plot 66 points, we must necessarily have at least 2 collisions, so $Pr(0, 66, 64) = 0$ and $Pr(1, 66, 64) = 0$.

The basic idea of the formula is to calculate, given p pixels and n plotted points, the ratio between the number of existing configurations with *exactly* k collisions and the number of possible total configurations.

$$Pr(k, n, p) = \frac{\#config\ with\ exactly\ k\ collisions}{\#total\ configurations}$$

The $\#\ of\ total\ configurations$ is computed considering all the possible ways to choose n points on an area of p pixels allowing collisions, i.e., selecting n elements from a set of p elements allowing repetitions (dispositions with repetitions: p^n).

Calculating the *# config with exactly k collisions* is performed in three steps.

1. Compute all the possible ways of selecting $n-k$ non colliding points from p pixels, that corresponds to selecting $n-k$ elements from a set of p elements without collisions, (combinations *without* repetitions), i.e., $\binom{p}{n-k}$.

2. For each of the above combinations, compute all the possible ways of hitting k times one or more of the $n-k$ non colliding points in order to obtain *exactly* k collisions, that corresponds to selecting k elements from a set of $n-k$ elements *with* repetitions (combinations *with* repetitions): $\binom{n-k+k-1}{k}$.

3. Step 2 computes *combinations*; since we are interested in *dispositions*, we calculate the permutations (PERM) of these combinations. Because of the variable number of duplicates (e.g., it is possible to have k collisions hitting k+1 times the same pixel p_i, or k times p_i and two times pixel p_j, or k-1 times p_i, two times pixel p_j, and two times pixel p_k and so on) it is not possible to express such permutations through a closed formula and we computed it through a C program.

From the above expressions we computationally derived a series of functions (see Figure 2) showing the behavior of the observed area as the number of plotted points increases. More precisely, on the Y axis we have:

- the mean of colliding elements k (as n percentage)
- the used space (as p percentage)
- the free space (as p percentage)

w.r.t. the number of plotted points n (X axis, as p percentage).

For example, if we have an area of 8×8 pixels ($p = 64$), the figure tells us that plotting 128 ($n = 128$) points (200% of p) we foresee an average of 72.5 (56.7% of n) collisions and, as a consequence, 8.5 free pixels ($d = 64-(128-72.5) = 8.5$) (13.4% of p). As the number of plotted points n increases, the percentage of collisions increases as well, while the free space decreases.

Using this graph we can derive several useful and objective indications on the degradation of an image. As an example, the graph tell us how much we are saturating the space or if the display is able to accurately represent

relative densities and, consequently, how much to sample the data to guarantee a prefixed visualization quality.

3.1 Data density and represented density

The previous results give us a way to measure and, consequently, control the number of colliding elements. Before describing quality metrics and optimization strategies, we need to clarify our scenario and to introduce new definitions. In particular, we need to differentiate the measurement of density in the data space from density in the device space.

We assume the image is displayed on a rectangular area 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 1280×1024 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 : *data density* 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 $0, \frac{1}{A}, \frac{2}{A}, \ldots, \frac{n}{A}$. For each distinct value we can count the number of sample areas characterized by that value, obtaining the data density distribution.

Represented density is defined as $RD_{i,j} = \frac{p_{i,j}}{A}$ where $p_{i,j}$ is the number of distinct active pixels in $SA_{i,j}$. 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.

Data densities are measured in a continuous space, while represented densities in a discrete one; because of collisions the number of active pixels on a sample area $SA_{i,j}$ will likely be less than the plotted points so $RD_{i,j} \leq D_{i,j}$.

4 The User Study

This section describes the studies we performed to understand what is the minimum difference in active pixels between two sample areas that allows for perceiving a density difference. Moreover, we investigated a second issue: does the distance between two different areas influence the users' density perception?

To answer these questions we performed two studies based on a comparison strategy [25], one investigating the first question, and the other one challenging the figures coming from the first experiment against the distance issue.

It is worth noting that our approach, even if based on dot-stimuli, differs from the ones discussing numerosity discrimination for three main aspects:

1. Area size. Our technique requires to investigate density perception in very little areas; typical numerosity discrimination studies use wider areas, spanning several degrees and, as a consequence, it is not immediate to apply the presented results to our environment (the least area considered while studying size invariance in [3] spans 40' of arc; a sample area of 8 by 8 pixels spans 13' of arc);

2. Dot nearness. We are considering small areas and pixels that very likely may touch, producing continuous patterns and saturation. Density and numerosity studies consider ideal dots without saturation and adjacency.

3. Exposure time. Most of the available studies involve a very short exposure time (typical values are less then a second) while the real usage of our visualization allows for a virtually infinite exposure time.

These differences pushed us to design an ad-hoc experiment, investigating on little areas with adjacent pixels. In order to reduce the experiment complexity we did not consider pattern issues and we used monochromatic pixels, neglecting the color influence on density perception [31]. As a consequence, while the experiment results fits quite well the objective of our proposal, generalize them is not a trivial task.

We involved 38 people in the user study (25 males and 13 females, ranging between 23 to 46) asking the subject needing glasses to wear them. According to the first objective, we asked the users to recognize few more dense areas on a uniform background (basis), repeating the test for different bases and different density differences. A typical experiment step is depicted on Figure 3. The image contains 100 sample areas, 97 of which filled randomly with the same number of active pixels (basis) while the remaining 3 are filled with extra pixels (δ). In the example the 97 sample areas are filled at 20% (basis=20) of their capacity and the 3 densest contain 150% more pixels than the basis ($\delta = 150$); in the figure the user identified and selected the uppermost densest sample area. A preliminary pilot study showed us that the δ values we were looking for depend on the basis and, in order to cope with this issue, we arranged the experiment as follows. All the users were presented (i.e., we performed a within subjects experiment) with 11 steps, corresponding to having the 97 equal sample areas progressively filled at 5%, 8%, 10%, 20%, ..., 90% of the sample area capacity. For each step, 5 substeps were performed, each of them showing 3 denser sample areas characterized by 5 increasing δ_i. The users were asked to select, for each substep, the 3 densest areas and the program recorded attempts and errors.

The results are collected in Figure 4. The two tables on the left show in each column a different basis $(5, \ldots, 90)$ expressed as percentage of active dots with respect to the capacity of the sample ares. Each row represents the incremental steps adopted in the the test. For each increment D the tables show the corresponding recognition percentage R expressed as percentage with respect to the

basis. As an example, the second column tells us that, while evaluating a basis of 10%, we asked the user to identify sample areas containing 55%, 65%, 75%, 85%, and 95% extra pixels and that the recognition rate was 62%, 77%, 82%, 92%, and 97%, respectively. The table on the right shows, for each basis, the increment that produced a successful recognition rate greater than 70 per cent. Linearly interpolating these values we derived a function $minimum\delta(RD_{i,j})$ returning the minimum increment a sample area must show to be perceived as denser than $SA_{i,j}$ (see the graph in the lower right of Figure 4).

The results of this first test were used as input for the second one. The users were presented with couples of sample areas ranging on the same 11 steps of the first experiment and differing in density exactly of $minimum\delta()$. The same couple was presented 5 times to the users in a random fashion and at variable distance and the users were asked, for each step, to select the denser area. A typical experiment step is depicted on Figure 5. We run an ANOVA test but we could not find any statistically significant difference, therefore, in our context, we consider distance as having no influence in the perception of density differences.

5 Quality metrics

In this section we provide several quality metrics. Some of them are intended for measuring the absolute image degradation, i.e., the metrics provide a way to evaluate the collision percentage, the fraction of the screen bearing not acceptable distortion, and the data percentage that is affected by visual degradation; other ones, computed through a weighted algorithm, focus on distorted areas and provide an indication on how many density differences are still visible in the displayed image. Moreover, since we sample the data that generates the image, we need to measure the negative effects of such an activity as well.

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 squared sample areas size in terms of number of pixels, l_pixels;
- the number of collisions k per sample area (SA) (as defined in Section 3);
- the data density and the represented density (as defined in Section 3.1).

The first quality metric we provide is the following:

$$\text{PPr(Points Pixels ratio)}=\frac{n}{x\_pixels \times y\_pixels}$$

that gives the measure of how much we are overplotting the screen.

This measure provides an overall information about the image degradation but it is not able to capture local distortions. As an example, if we apply it to Figure 1, displayed on a 600×600 pixels area, we obtain quite a good value, 0.43, denoting that pixels are more than twice the plotted points. The problem is that overplotting is associated with a very little area; the following metrics are thought to discover such kinds of distortions. First of all we want to measure the percentage of distorted points:

$$\text{CPr(Collisions Points ratio)}=\frac{k}{n}$$

that gives the measure of the overall collisions/points ratio.

Still referring to Figure 1, such a metric is equal to 0.73: roughly speaking we can say that 73% of the data set is colliding. What we still miss is the information about the screen percentage bearing such a distortion. To this aim, 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. $l\_pixels \times l\_pixels$ (that, according to Figure 2 corresponds to an overplotting of 161 %). Obviously,

the lower this value the better the image and Δ is a parameter that allows for fine-tuning the algorithms described in Section 6.

Using Δ we can define the following metric:

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

that gives the measure of the screen percentage affected by a non acceptable distortion; in our example BGSAr is equal to 4% (the area very close to the axes origin). In order to measure the data percentage belonging to such a distorted area we define the following metric:

$$\text{CPPr(Crowded Points Points ratio)}=\frac{\#\ of\ points\ falling\ in\ a\ SA\ showing\ k>\Delta}{n}$$

CPPr in our example is equal to 70%, a very bad value. Combining the last two metrics we can say that 70% of the data set is represented in a very small (4%) and crowded area.

Summarizing, we can say that while the screen dimension should nicely bear the image (PPr=0.43) we are experimenting a very high number of collision (CPr=0.73) that are concentrated in a very small screen area (BGSAr=0.04) and that most of the data points (CPPr=0.70) are not adequately represented (w.r.t. $\Delta = 32\%$).

Till now we focused on collisions and distorted areas; now we concentrate on relative densities, measuring the lost density differences through the metric LDDr (Lost Data Densities ratio). This metric is calculated comparing couples of sample areas and checking whether their relative data density (D) is preserved or not 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 $Diff(D_{i,j}, D_{k,l}) = Diff(RD_{i,j}, RD_{k,l})$.

To produce a measure, we apply an algorithm that considers all the couples of Distorted SAs (DSA), comparing their D and RD through the $Diff$ function and counting the non matching pairs [2].

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:

```
function LDDr(){
 Let DSA[m][r];  \* distorted sample areas
 Let couples=0;  \* weighted couples of distinct distorted sample areas
 Let sum=0;      \* weighted non matching couples of distinct distorted sample areas
 foreach distinct pair(DSA[i][j], DSA[k][l]){
   couples = couples + pt(DSA[i][j]) + pt(DSA[k][l]);
   if ( NOT match(i, j, k, l) )
      sum = sum + pt(DSA[i][j])+ pt(DSA[k][l]);}
 return (sum / couples);}
```

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

The variable *sum* contains the number of weighted non matching couples encountered during the iterations; dividing it by the weighted total number of possible distinct distorted couples we obtain the weighted percentage of non matching sample areas ranging between 0 and 1 (the lower the better).

The main drawback of this metric is that it uses numerical differences between sample areas to decide whether a data density difference is well represented or not 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.

[2] We count non matching pairs to produce a metric that behaves as the other ones: high values of the metric correspond to a bad situation

The function $minimum\delta()$ defined in the end of Section 4 allows for defining 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, we obtain the PLDDr (*Perceptually* Lost Data Densities ratio) metric. In this way the quality metric deals with user perceptible vs numeric density differences.

In order to better understand the difference between the two metrics, we apply them against the example in Figure 1. Using the pure numeric metric, LDDr (with $\Delta = 32\%$), we obtain the reasonable value of 0.29, 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, by contrast, the PLDDr metric we obtain a worse (but more realistic) value, 0.57, meaning that in the distorted area only 43% of the data points are presented correctly. That implies that the pure numeric metric counts a great number of "fake" density differences not perceivable by the users.

Finally, we have to consider the negative effects of the sampling activity; through the following metric we measure the portion of the screen that has been emptied by the sampling.

$$\text{ESAr (Erased SAs ratio)} = \frac{\#emptySA\ after\ sampling - \#emptySA\ before\ sampling}{\#\ of\ SA}$$

The above metrics give some precise clues about the image degradation and, in the following, we provide some general examples of how to exploit them; more formal usage of such metrics is in Section 6:

1. given a data set and a prefixed display area size, it is possible to check the quality figures against a predefined set of threshold values and, in case of

violation, to sample the data until all the threshold are satisfied (details about how and how much to sample are provided in Section 6);

2. given a data set and a set of threshold values, it is possible to compute the minimum display area size preserving all the threshold values, allowing the system to optimize the screen usage;

3. given a data set and a prefixed display area size, it is possible to devise a non monotonic quality index, i.e., a linear combination of two or more contrasting quality indexes (e.g., ESAr and BGSAr) or an inherently non monotonic quality index (e.g., PLDDr) and to choose the sample ratio that minimizes such a quality index.

6 Sampling the data set

In this section we introduce two sampling strategies, namely *uniform sampling* and *non uniform sampling*. We apply them against the metrics introduced in Section 5. In the following examples we use the monotonic CPr and the non monotonic PLDDr metrics as representative cases; the same approach can be applied considering the other quality metrics.

6.1 Uniform sampling

Uniform sampling presents two main advantages: it is easy to implement and preserves the intensity of density differences; on the other hand, to reveal density differences in very crowded zones requires a sample ratio that destroys data in faint areas.

Using the quality metrics introduced in Section 5 we can measure, for a given sampling factor, the quality of the generated image and than we can estimate the minimum amount of sampling to apply to produce an ideal final representation. For single, monotonic metrics the approach is quite straightforward: we choose

a quality threshold and we sample the image as much as needed to satisfy it. As an example, using the image shown in Figure 1, displayed on a 600×600 pixels screen, assume that we want to generate a new image presenting a $CPr \leq 0.6$ (against the initial value 0.73). Applying decreasing *sampling factors* we find that 40% is the first value that satisfies the constraint (CPr=0.598).

Conversely, we may decide to not sample the data and to enlarge the image to 976×976 pixels (CPr=0.597). Such examples clarify the first two ways of exploiting the metrics discussed at the end of Section 5.

If we are dealing with non monotonic metrics (e.g., PLDDr), or with a linear combination of metrics showing opposite behavior (e.g., ESAr and BGSAr), we have to follow a different approach, looking for the sampling factor that minimizes the function. As an example, still referring to data on Figure 1, let us assume that we want to minimize the PLDDr metric that, for the original image is equal to 0.43. If we apply ours algorithm we obtain the optimal sample ratio of 22% and the PLDDr metric reaches the minimum value of 0.19.

6.2 Non uniform sampling

Applying the same amount of sampling to the whole image is straightforward but presents several drawbacks. For instance, to sample areas presenting very low data density is useless and potentially dangerous, because empty areas may appear where data was previously plotted [3]. In addition, in some cases the user is interested in discovering as many density differences as possible, neglecting their intensity. Therefore, we introduce a novel non uniform sampling approach able to (a) preserve low density areas and (b) to show to the user a greater number of density differences.

The main idea is to apply a different sample ratio to each sample area, to obtain a suitable represented density distribution. In fact, the problem of

[3]This problem is captured by the ESAr metric that can be combined with other metrics to find out optimal uniform sampling values

preserving relative densities can be challenged altering the mapping between the set of the actual data densities and the set of available represented densities; while the uniform sampling approach alters the mapping in linear way, non uniform sampling forces different data densities to be represented on the *same* represented density, through different sampling ratios. Moreover, because of numerical differences are not always perceived, as discussed in Section 4, we need to fine tune our technique exploiting the user study results.

In the following we use an artificial numeric example to explain our approach; next section will provide a real example. For the sake of the clarity, we present our technique in two steps: first we describe our algorithm using numerical differences and, after that, we introduce the perceptual modifications.

6.2.1 The numerical algorithm

Let us assume we are plotting points on a 40×40 pixel screen arranged in 100 4×4 sample areas. In the example we concentrate on the number of data elements or active pixels neglecting the SA area (what we called A), that is just a constant. In Figure 6(a) the data densities (in terms of number of points) corresponding to each sample area are displayed.

Figure 6(b) shows the data density distribution; as an example, we can see that the maximum data density 56 is shared by five sample areas. Figure 6(c), obtained applying the statistical results discussed in Section 3, shows the actual represented density (in terms of active pixels) ranging, for each $SA_{i,j}$, between 1 and 16; figure 6 (d) shows the represented density distribution.

Comparing the data densities with the represented densities we discover that 41% of the visualization pane ranging between 30 and 56 data density collapsed on just three different represented data densities (14, 15, and 16).

In order to improve such a situation we compute a new mapping among the existing data densities and the 16 available represented densities.

To obtain such a result, starting from the data density distribution (Figure 6

(b)), we split the X axis in 16 (i.e., the available represented densities) intervals, each of them containing the same number of SAs, i.e., 6.25=(100/16). Because we are working on discrete values, we cannot guarantee that the average value is 6.25 and, as a consequence, we use an algorithm that minimizes the variance. After that, the SAs belonging to interval i are sampled to produce a represented density equal to i, as depicted on Figure 7. In the example, the first interval encompasses 6 SAs with data density 1 and no sampling is required. The second interval encompasses 3 SAs with data densities 2, 1 SAs with data densities 3, and 4 SAs with data densities 4; they are sampled as much as needed to produce a represented density equal to 2. The third interval encompasses SAs with data densities 5, 6, and 7 that are sampled as much as needed to produce data density equal to 3, and so on.

The represented densities resulting from this approach are depicted in Figure 8 (a); Figure 8 (b) shows the new, more uniform density distribution. In this new representation the 41 % data densities that collapsed on just three different values (14, 15, and 16) now span on the interval between 10 and 16. On the other hand, as an example of the distortion introduced by the method, the difference between data densities 25 and 11 (2.27), previously mapped on represented densities 8 and 13 (1.62) is now mapped on 5 and 8 (1.60).

6.2.2 Perceptual issues

We can say that the algorithm tries to maximize the number of density differences through a uniform represented density distribution. But, as pointed out in Section 4, this trail of thoughts does not take into account that numerical differences below a certain threshold are not perceived, so it make no sense to use *all* the possible represented densities. For a given sample area size, using the results presented in Section 4, it is possible to compute the largest ordered subset of represented densities values that are perceived as different by users. We call them *perceptual densities*.

As a numerical example, considering the more realistic case of 8×8 pixels SAs used in the next section, we devised the figures depicted on Table 1. Starting from the first perceptual density 1 we computed, using the $minimum\delta()$ function, the second perceptual density: $1 + minimum\delta(1) = 2$; in a similar way we computed the third one as $2 + minimum\delta(2) = 4$, and so on. The table shows, on the right part, the 14 perceptual densities; for each of them the left part groups the represented densities that collapse on the perceptual ones. According to this result, the algorithm described so far proceeds taking into account *only* the perceptual densities while splitting the data density distribution axis. That corresponds, in the case of 8×8 pixels SAs used in the following examples, to consider only 14 intervals instead of 64 and sampling the SAs belonging to the interval i in order to produce a represented density equal to the i_{th} perceptual density. As an example, all the SAs belonging to the 5_{th} interval are sampled to produce a represented density equals to 11.

Roughly speaking, we can think of the whole process as follows. We have at disposal p different represented densities that are matched against k 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, the original data densities, altering their assignment to the p represented densities to maximize the number of correctly represented density differences. Moreover, since the problem of perceptual differences is recognized, we apply the algorithm not to all the p represented densities but to the subset of them that produces the maximum number of perceivable density differences, i.e., the perceptual densities.

6.3 An example with a real dataset

In this section we apply our techniques against the dataset used in Section 6.1: the one containing 160,000 mail parcels (displayed on a 304×304 screen and

using 8×8 pixels SAs). Note that the images used in this section have the same dimension of the screen shots they come from, in order to preserve, as much as possible, the original visual feeling.

Figure 9 shows the original, crowded, visualization (no sampling) while Figure 10 shows the same image enlarged enough to show most of the details of the crowded area. More precisely, the original image is characterized by the following figures: CPr=0.80, BGSAr=0.149, and CPPr=0.79 while the enlarged one presents the following values: CPr=0.51, BGSAr=0.01, and CPPr=0.34. Note that the enlarged image is provided for reference purposes: it shows the data insights that the sampling algorithms should reveal.

Figure 11 shows the results of the best uniform sampling algorithm minimizing the PLDDr metrics: the algorithm discovered a minimum with a sampling factor of 22%. The sampling makes evident the clusters close to the origin and the clusters very close to the X axis. On the other hand, the faint zones in the uppermost part of the images are badly represented.

Conversely, Figure 12 shows the result obtained applying the perceptual non-uniform sampling algorithm. High density areas present more density differences than both the original image and the one obtained with the best uniform sampling. The PLDDr metric, measuring the perceptually lost density differences, confirms the visual impression: it is equal to 0.43 against the original value 0.63 and the one obtained using the best uniform sampling algorithm 0.51. Moreover, the algorithm does not alter low density areas that are represented as they appear on the original image. We can say that the non uniform sampling produces the advantages of both strong and weak sampling: crowded areas are sampled enough to show interesting patterns, as happens with strong sampling, faint areas remain quite untouched, as happens with weak sampling.

7 Working prototype and implementation

To validate our ideas and tune up the parameters of our methods we developed an analysis and inspection tool that permits to calculate metrics and to apply sampling algorithms. The basic functionalities of the system are:

- *Quality metrics computation* - The user can inspect a list of quality metrics associated with the current visualization;
- *Sampling algorithms execution* - The user can run both uniform and non-uniform sampling algorithms. While using uniform sampling, s/he can either interactively sample the image or set some metric thresholds forcing the systems to return the sampled image satisfying them;
- *Interactive image inspection* - It is possible to use dynamic filters on sample areas. For instance, it is possible to interactively filter out or brush sample areas with number of values above/below a given threshold.

In case of interactive activities, the system memorizes the intermediate results reusing them in order to save time and avoiding pixel flickering. As an example, the system allows for manually sampling the data at different ratio (1..99%). This is obtained assigning, once for all, to each data element a random integer ranging between 1 and 100. Sampling to 23% is obtained discarding all the elements whose associated integer is greater than 23. In this way, if the user browses different sampling ratios, redisplaying the same sampling percentage more times, s/he will be presented with continuously and homogenously changing images.

The sampling algorithms work in a one-shot fashion, switching back and forth between the original image and the sampled one. In order to maintain visual continuity, data samples are collected only once when the algorithm runs for the first time. When the user swap to the original visualization and back again to the sampled one, the representation stays the same, thus achieving

visual consistency. A specific function is provided to explicitly request for a new "fresh sample" when needed.

These implementation details are relevant when using the system for exploration activities. It is worth to note, in fact, that our methods can be used to detect potentially interesting patterns by moving back and fort from the original and the final image, and changing the available settings (we often informally noticed this kind of patterns while using the tool) in an explorative fashion. When this happens, visual continuity and consistency obtained through the solutions outlined above, become crucial.

8 Extensions of our approach

The work presented in this paper deals with 2D scatter plots using monochromatic points; it is our opinion, however, that it is possible to extend our approach to other Infovis techniques (e.g., scatter plots using colors and shapes, parallel coordinates, etc.); however, such an activity is not a trivial task and the aim of this section is to describe the overall methodology and the involved challenges.

Extending our approach can be done pursuing two different methods. The first one, useful when the target visual representation is very far from scatter plots, is to devise a bidirectional mapping between the visual representation and a scatter plot: first we transform the representation in a scatter plot, after that we apply the sampling algorithms, and finally we step back to the original visualization. We applied this strategy to parallel coordinates, obtaining quite encouraging results (see, e.g.,[6]); however the overall approach is not mature enough and it is out of the scope of this paper to discuss it. The second method is based on the idea of redefining our concepts (data density, represented density, density estimation, etc.) in term of the new target visualization, applying our algorithms directly on it. Again, our results are very preliminary; however, in our opinion, it is interesting to provide an overview of the problems and the

challenges we are dealing with while trying to extend our approach to 2D scatter plots using shapes and colored dots.

- Sample area size. While the use of color does not affect this issue, the size of the sample areas must be increased according to the shapes' dimension. It is not clear (1) how much and (2) what is the effect of this new size on the uniform distribution assumption.

- Collisions. Both colors and shapes require a new way of defining and handling collisions. Concerning colors, if the colliding points are of different colors a strategy for assigning the resulting color is is needed. A discussion of the matter is in [17] and a reasonable choice is to select the color that occurs most frequently in the collision. Considering shapes, the definition of collision is based on the idea of computing the percentage of overlapping area between two shapes and the resulting notion of collision can be either binary or fuzzy. In the first case, a collision happens when the overlapping percentage is more than a threshold; obviously, the threshold value is a critical parameter and its definition requires a non trivial analysis, involving user studies. On the other hand, it is possible to weight each collision with a value corresponding to the normalized value of the overlapping areas. The way of considering these different collisions while computing the quality figures is, again, a non trivial task.

- Data and Represented density definition. For colored points both data and represented density require to take into account the color distribution; moreover, because of density perception is strongly affected by the involved colors [31], user studies and perceptual metrics require a new, non trivial, definition. Considering shapes, a simple approach foresees to apply the same definitions provided for points considering the shapes's barycenters falling within a sample area. A more precise calculation is based on the idea of computing, for each shape, the area percentage falling within a

sample area.

- Density estimation. The probabilistic formulae must be (non trivially) revised considering color distribution and that, in case of shapes, we have to take into account collisions belonging to a sample area generated by adjacent sample areas.

9 Conclusion And Future Work

In this paper we presented two complementary combinatorial sampling techniques, uniform and non uniform sampling, that aim at automatically dealing with overplotting in 2D scatter plots. The techniques exploit some statistical results and a formal model describing and measuring overplotting, screen occupation, and both data density and represented density. Such a model allows for defining precise and sound quality metrics that are used for: (a) measuring in an objective way the degradation of several data characteristics and (b) computing the right amounts of sampling to apply in order to guarantee some quality parameters. The overall formal framework takes into account the results of ad-hoc user studies providing precise figures about the perception of density differences.

We are actually trying to extend our approach to other Infovis techniques; the issues rising from such an activity have been described in Section 8.

10 Acknowledgements

We would like to thank Pasquale Di Tucci and Francesco Degrazia for their invaluable help in implementing the software prototype. Morever, we thank the anonymous referees for the valuable comments and suggestions that contributed to improve the quality of the paper. Work supported by the MIUR-FIRB project "MAIS" (Multichannel adaptive Information Systems, http://www.mais-project.it/)

References

[1] H.B. Barlow A. Burgess. The precision of numerosity discrimination in arrays of random dots. *Vision Research*, 23(8), 1983.

[2] 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.

[3] J. Allik, T. Tuulmets, and P.G. Vos. Size invariance in visual number discrimination. *Psychological Research*, 53(4), 1991.

[4] Almir Olivette Artero, Maria Cristina Ferreira de Oliveira, and Haim Levkowitz. Uncovering clusters in crowded parallel coordinates visualizations. In *Proc. of IEEE Symposium on Information Visualization*, pages 81–88. IEEE Computer Society, 2004.

[5] Barry G. Becker. Volume rendering for relational data. In *Proc. of the IEEE Symposium on Information Visualization*, page 87, Washington, DC, USA, 1997. IEEE Computer Society.

[6] Enrico Bertini, Luigi Dell'Aquila, and Giuseppe Santucci. Reducing infovis cluttering through sampling displacement and user perception. In *Proc. of IEEE Symposium on Information Visualisation Infovis05*, October 2005.

[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] Geoff Ellis and Alan Dix. Density control through random sampling: an architectural perspective. In *Proc. of IEEE Conference on Information Visualisation*, pages 82–90, July 2002.

[9] M. A. Fisherkeller, J. H. Friedman, and J. W. Tukey. *Dynamic Graphics for Statistics*, chapter PRIM9: An Interactive Multidimensional Data Display and Analysis System, pages 140–145. Wadsworth, Inc., 1988.

[10] 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.

[11] Ying-Huey Fua, Matthew O. Ward, and Elke A. Rundensteiner. Navigating hierarchies with structure-based brushes. In *Proc. of the 1999 IEEE Symposium on Information Visualization*, page 58. IEEE Computer Society, 1999.

[12] George W. Furnas. Generalized fisheye views. In *Proc. of ACM SGCHI Conference on Human factors in Computing Systems*, pages 16–23. ACM Press, 1986.

[13] N. Ginsburg and A. Nicholls. Perceived numerosity as a function of item size. *Perceptual and Motor Skills*, 67(2), 1988.

[14] Georges G. Grinstein, Patrick E. Hoffman, and Ronald M. Pickett. Benchmark development for the evaluation of visualization for data mining. pages 129–176, 2002.

[15] Kasper Hornbæk, Benjamin B. Bederson, and Catherine Plaisant. Navigation patterns and usability of zoomable user interfaces with and without an overview. *ACM Trans. Comput.-Hum. Interact.*, 9(4):362–389, 2002.

[16] Alfred Inselberg and Bernard Dimsdale. Parallel coordinates: a tool for visualizing multi-dimensional geometry. In *Proc. of Visualization '90*, pages 361–378. IEEE Computer Society Press, 1990.

[17] Dean F. Jerding and John T. Stasko. The information mural: A technique for displaying and navigating large information spaces. *IEEE Transactions on Visualization and Computer Graphics*, 4(3):257–271, 1998.

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

[19] 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.

[20] John Lamping and Ramana Rao. Visualizing large trees using the hyperbolic browser. In *Proc. of ACM SGCHI Conference on Human factors in Computing Systems*, pages 388–389. ACM Press, 1996.

[21] Ying K. Leung and Mark D. Apperley. A review and taxonomy of distortion-oriented presentation techniques. *ACM Transactions on Computer-Human Interaction*, 1(2):126–160, 1994.

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

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

[24] Brath Richard. Concept demonstration: Metrics for effective information visualization. In *Proc. of IEEE Symposium on Information Visualization*, pages 108–111. IEEE Service Center, Phoenix, AZ, 1997.

[25] F. Vardabasso S. Alam, R. Luccio. Regularity, exposure time and perception of numerosity. *Perceptual and motor skills*, 63(2), 1986.

[26] T. C. Sprenger, R. Brunella, and M. H. Gross. H-blob: a hierarchical visual clustering method using implicit surfaces. In *Proc. of Visualization '00*, pages 61–68. IEEE Computer Society Press, 2000.

[27] S. S. Stevens. *Sensory Communication*, chapter The Psychophysics of sensory functions, pages 1–33. MIT Press, 1961.

[28] Marjan Trutschl, Georges G. Grinstein, and Urska Cvek. Intelligently resolving point occlusion. In *Proc. of IEEE Symposium on Information Visualization*, 2003.

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

[30] P.G. Vos, M.P. Oeffelen, H.J. Tibosch, and J. Allik. Interactions between area and numerosity. *Psychological Research*, 50(3), 1988.

[31] K.T. Mullen W.H.A. Beaudot. Role of chromaticity, contrast, and local orientation cues in the perception of density. *Perception*, 29(5), 2000.

[32] Graham J. Wills. An interactive view for hierarchical clustering. In *Proc. of IEEE Symposium on Information Visualization*, page 26. IEEE Computer Society, 1998.

[33] Allison Woodruff, James Landay, and Michael Stonebraker. Constant density visualizations of non-uniform distributions of data. In *Proc. of ACM Symposium on User Interface Software and Technology*, pages 19–28. ACM Press, 1998.

[34] Allison Woodruff, James Landay, and Michael Stonebraker. Vida: (visual information density adjuster). In *Extended Abstracts of ACM SGCHI Con-*

ference on Human factors in Computing Systems, pages 19–20. ACM Press, 1999.

[35] Li Yang. Interactive exploration of very large relational datasets through 3d dynamic projections. In *KDD '00: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 236–243, New York, NY, USA, 2000. ACM Press.

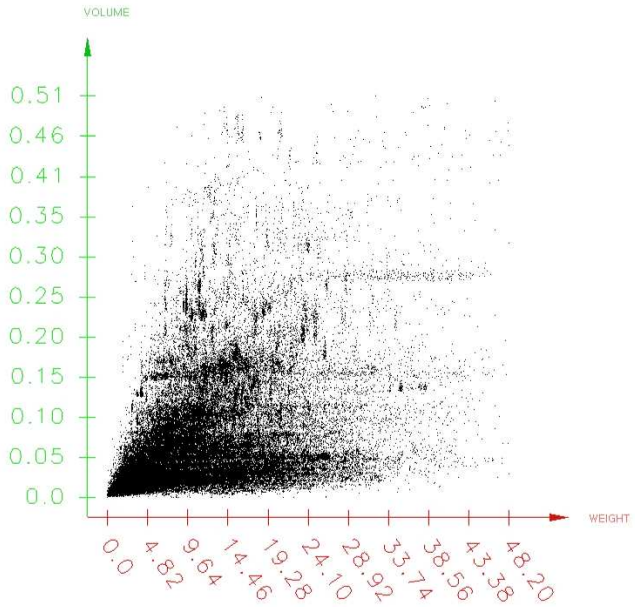


Figure 1: Plotting mail parcels: an example of a visualization with a high number of overlapping items.

Rep Density	Perc 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.

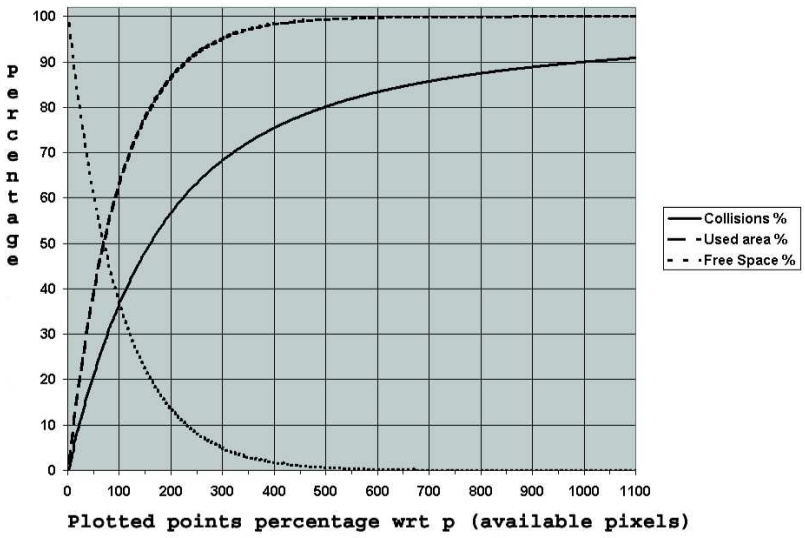


Figure 2: Plotted functions showing the behavior of overplotting as the number of plotted points increases.

Figure 3: Testing numerical differences issues

Basis = 5		Basis = 8		Basis = 10		Basis = 20		Basis = 30		Basis = 40	
D	R	D	R	D	R	D	R	D	R	D	R
70	53	60	42	55	62	35	41	30	62	26	67
90	73	80	69	65	77	40	64	35	56	30	77
110	89	100	84	75	82	45	70	40	74	34	85
130	93	120	93	85	92	50	77	45	95	38	90
150	98	140	95.5	95	97	55	87	40	97	42	100

Basis = 50		Basis = 60		Basis = 70		Basis = 80		Basis = 90	
D	R	D	R	D	R	D	R	D	R
20	59	22	72	12	64	10	70	6	77
22.5	79	25	92	13.5	67	11.5	87	7	92
25	77	28	100	15	73	13	95	8	100
27.5	92	31	97	16.5	77	14.5	97	9	97
30	95	34	100	18	90	16	97	10	100

D = percentual increment in data points with respect to basis
R = percentage of succesful recognitions of density differences

Basis	DT(%)	DT(#)	DT(#)/B(#)
5	87	2,78	0,36
8	81	4,15	0,33
10	60	3,84	0,25
20	45	5,76	0,18
30	39	7,49	0,16
40	27	6,91	0,11
50	21	6,72	0,09
60	22	8,45	0,09
70	14	6,27	0,06
80	10	5,12	0,04
90	6	3,46	0,02

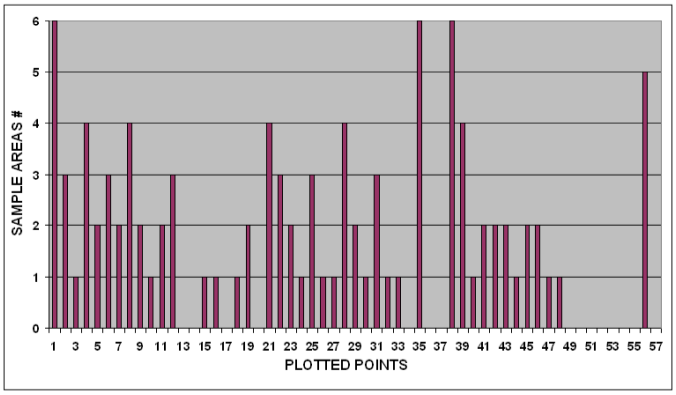
DT(%) = incremental threshold (% w.r.t. basis)
DT(#) = incremental threshold (num. of dots)
B(#) = number of dots in the basis

Figure 4: The user study results (all values are percentages).

Figure 5: Testing distance issues

46	1	11	25	1	31	19	21	18	1
4	3	12	38	45	56	26	35	12	38
38	38	35	56	28	8	21	42	22	7
35	24	4	1	30	9	41	31	21	23
41	47	45	6	35	22	7	56	38	28
8	4	39	56	28	27	42	22	43	29
1	35	21	39	44	35	1	6	5	25
32	2	31	29	8	28	33	39	5	40
43	4	12	16	23	2	9	48	39	8
11	25	56	46	10	38	2	15	6	19

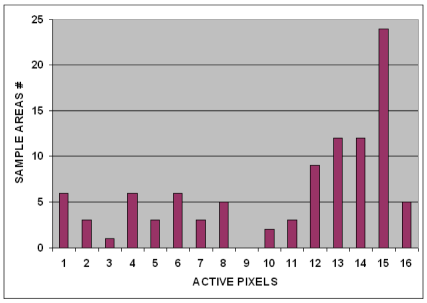
(a)



(b)

15	1	8	13	1	14	11	12	11	1
4	3	8	15	15	16	13	14	8	15
15	15	14	16	13	6	12	15	12	6
14	13	4	1	14	7	15	14	12	12
15	15	15	5	14	12	6	16	15	13
6	4	15	16	13	13	15	12	15	13
1	14	12	15	15	14	1	5	4	13
14	2	14	13	6	13	14	15	4	15
15	4	8	10	12	2	7	15	15	6
8	13	16	15	7	15	2	10	5	11

(c)



(d)

Figure 6: A screen area made of 100 sample areas: data density (a, b) and represented (c, d) density.

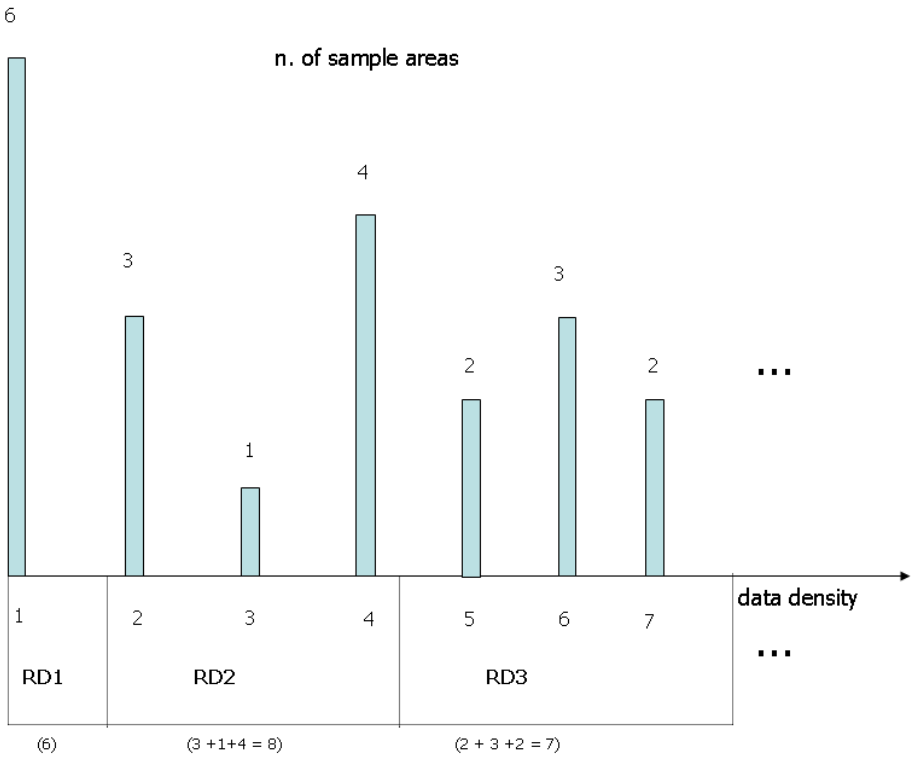
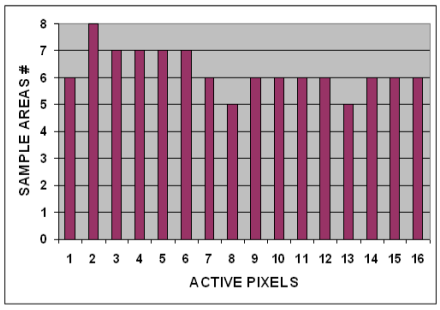


Figure 7: Data density distribution split algorithm

15	1	5	8	1	10	6	6	6	1
2	2	5	12	15	16	8	11	5	12
12	12	11	16	9	4	6	14	7	3
11	7	2	1	10	4	14	10	6	7
14	15	15	3	11	7	3	16	12	9
4	2	13	16	9	8	14	7	14	9
1	11	6	13	15	11	1	3	3	8
10	2	10	9	4	9	10	13	3	13
14	2	5	5	7	2	4	16	13	4
5	8	16	15	4	12	2	5	3	6

(a)



(b)

Figure 8: The distribution of represented density obtained by applying non uniform sampling.

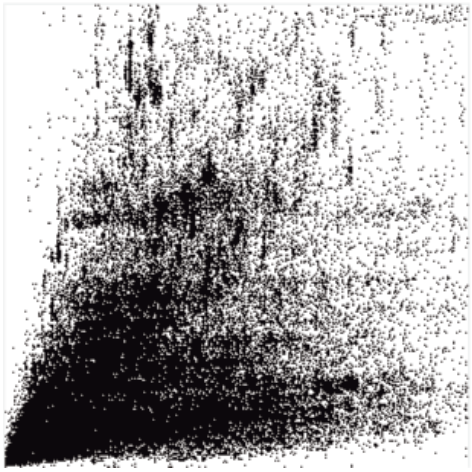


Figure 9: Original image (304 x 304 pixels)

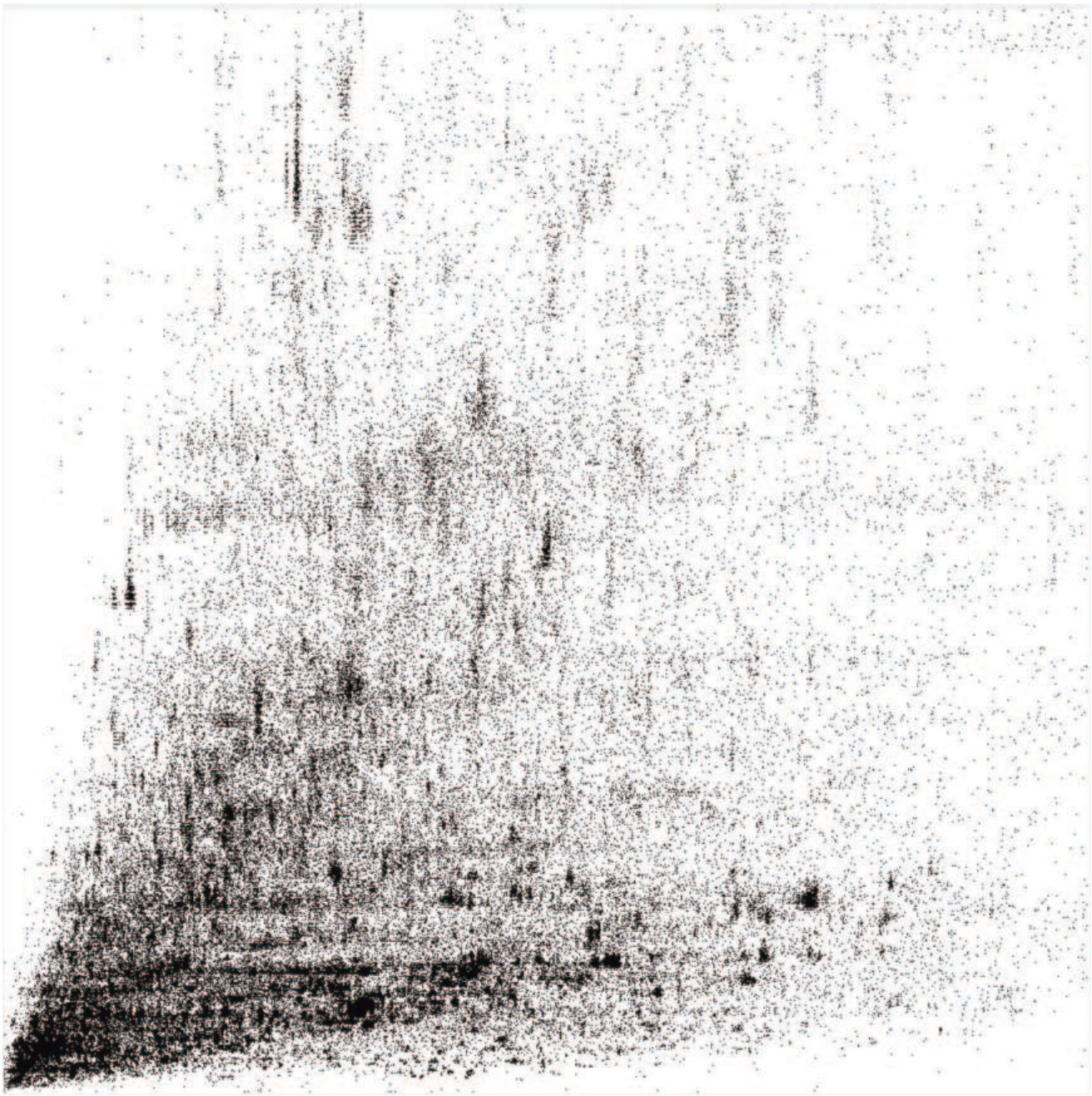
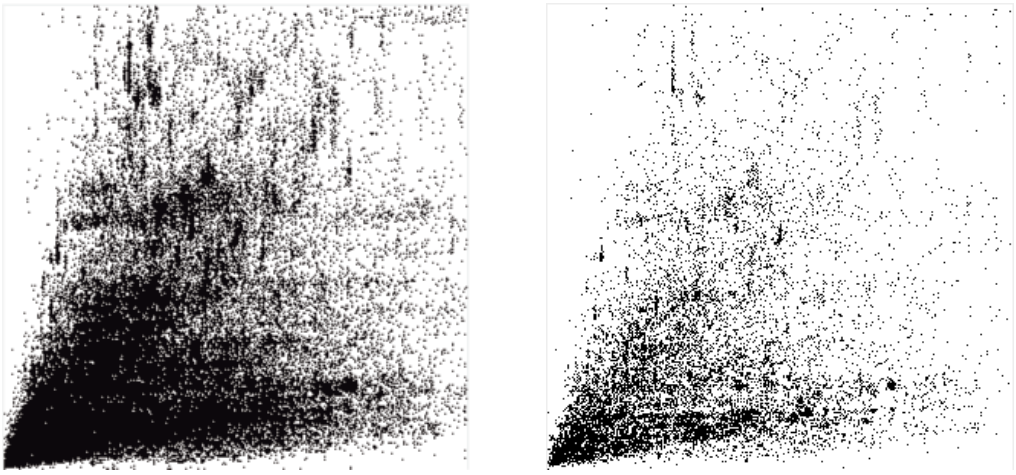
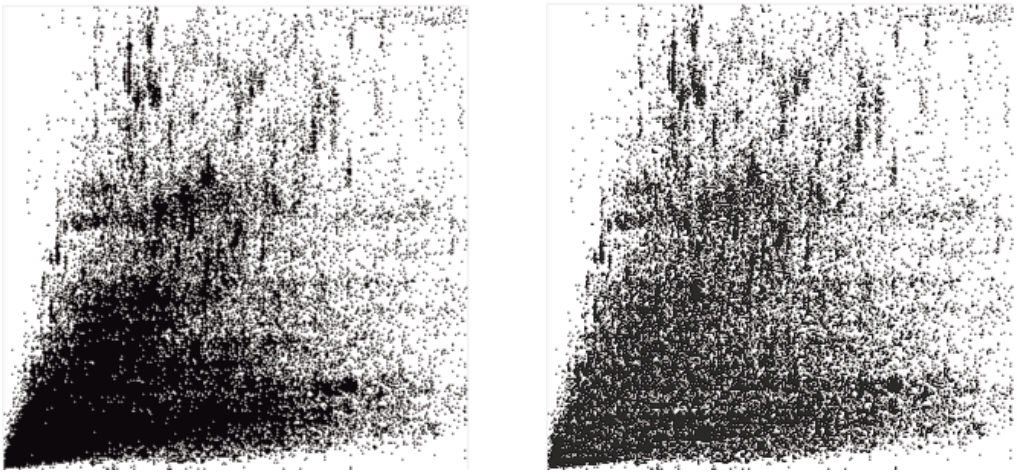


Figure 10: Enlarged original image (864 x 864 pixels)



(a) (b)

Figure 11: Uniform sampling: a) original image , b) best uniform sampling .



(a) (b)

Figure 12: Non uniform sampling: a) original image , b) perceptual non uniform sampling.