# Visuelle Datenanalyse mit Streudiagrammen: Problemlösungen für Ungleichverteilung und Überdeckung

**Visual Analytics with Scatter Plots:**
**Coping with Unequal Distribution and Overplotting**

Master-Arbeit an der Universität Konstanz im Fachbereich Informatik und Informationswissenschaft

vorgelegt von

Halldór Janetzko

Erste korrigierte Fassung vom 8.10.2010

| | |
|---|---|
| Einreichung: | 2. Juli 2010 |
| Prüfer: | Prof. Dr. Daniel A. Keim, Universität Konstanz |
| | Prof. Dr. Oliver Deussen, Universität Konstanz |
| Betreuer: | Dr. Peter Bak |

# Zusammenfassung

Streudiagramme gehören zu den mächtigsten und vielseitigsten Techniken, welche häufig im Bereich der visuellen Datenanalyse verwendet werden. Ein schon lange bekanntes Problem der Streudiagramme ist, dass diese häufig einen hohen Grad an Punktüberdeckungen enthalten. Dabei können signifikante Teile der Daten verdeckt werden, was die visuelle Datenanalyse erschwert. Zusätzlich behindert eine Ungleichverteilung der Daten die sinnvolle Darstellung mittels Streudiagrammen. Diese Masterarbeit befasst sich mit der Entwicklung eines neuen Ansatzes, den Generalized Scatter Plots, welche die überdeckungsfreie Visualisierung großer Datenmengen ermöglicht. Die grundlegende Idee ist es, dem Benutzer eine freie Wahl des Verzerrungsgrades und der Menge an erlaubter Überdeckung anzubieten, um die bestmögliche Ansicht zu generieren. Hierbei kann zwischen dem traditionellen Streudiagramm und der hier vorgestellten Technik kontinuierlich interpoliert werden. Zudem wird eine Optimierungsfunktion aufgestellt, welche sowohl Verzerrung als auch Überdeckung berücksichtigt. Außerdem werden die Generalized Scatter Plots auf einige Datensätze aus der realen Welt angewendet. Unter anderem werden dabei Daten aus den Anwendunsgebieten der Serverperformanz, der Telelefonnutzung und der geographisch bezogenen Einkommensstatistik verwendet. Der Vergleich mit anderen schon bestehenden Techniken zeigt abschließend die Vorteile der hier vorgestellten Technik.

# Abstract

Scatter plots are one of the most powerful and most widely used techniques for visual data exploration in order to detect patterns and correlations. A well-known problem is that scatter plots often have a high degree of overlap, which may occlude a significant portion of the data values shown. Additionally, scatter plots suffer from unequal data distribution, because dense areas are not visualized as good as sparse areas. The research shown in this master's thesis will cope with these problems using a novel approach called Generalized Scatter Plot. This technique allows an overlap-free representation of large datasets to fit entirely into the display. The basic idea is to allow the analyst to optimize the degree of overlap and distortion to generate the best possible view. To allow an effective usage, the capability to interpolate smoothly between the traditional and the generalized scatter plots is provided. In particular, an optimization function will be identified, which takes both overlap and distortion of the visualization into account. Furthermore, the generalized scatter plots will be applied to a number of real-world data sets from application domains, such as server performance monitoring, telephone service usage analysis, and geographical data, demonstrating the benefits of the generalized scatter plots over traditional techniques.

# Inhaltsverzeichnis

# Abbildungsverzeichnis

# 1 Einführung

Die ersten Schritte einer Datenanalyse laufen meist darauf hinaus, Einblick in die Daten zu erhalten und mittels einer ersten visuellen Analyse die Daten zu verstehen. Um beispielsweise Einblick in einen zweidimensionalen Datensatz zu erhalten, sind Streudiagramme das Mittel der Wahl. Streudiagramme haben den Vorteil, eine der verständlichsten und intuitivsten Darstellungen von zweidimensionalen Daten zu sein. Jeder Schüler hat schon im Mathematikunterricht Punkte oder Geraden in ein kartesisches Koordinatensystem eingetragen. Daher sind Streudiagramme, bei denen einfach nur Punkte in ein Koordinatensystem eingezeichnet werden, eine intuitiv verständliche Visualisierung.

Ferner sind Streudiagramme sowohl in Statistikprogrammen (zum Beispiel R [10]) als auch in Data Mining Tools (beispielsweise KNIME [12]) schon vorhanden und können daher einfach eingesetzt werden. Zudem kann das Streudiagramm als eine der Basistechniken angesehen werden, welche ein sehr breites Einsatzspektrum besitzt und extrem wandelbar und flexibel ist:

> „Indeed, among all the forms of statistical graphics, the humble scatterplot may be considered the most versatile, polymorphic, and generally useful invention in the entire history of statistical graphics.“ [11]

Es gibt viele Anwendungsdomänen, in denen große Mengen von mehrdimensionalen Daten anfallen. Beispielsweise treten diese Daten bei Telefondiensten, im Finanzsektor und beim Monitoring von Servern auf, um nur einige wenige Anwendungsgebiete zu nennen. Datenanalysten sind hierbei besonders daran interessiert, wie sehr sich ein Attribut auf die Ausprägungen eines anderen Attributs auswirkt. Schon 1984 haben William Cleveland et al. die herausragende Stellung des Streudiagramms bei der Datenanalyse hervorgehoben:

> „The scatter plot is one of our most powerful tools for data analysis.“ [8]

Heutzutage sind Streudiagramme immer noch eine der am häufigsten verwendeten Techniken, um mehrdimensionale Daten darzustellen. Durch die Visualisierung von Daten mittels Streudiagrammen können Beziehungen zwischen zwei Attributen, Cluster von Punkten und auch Ausreißer identifiziert werden. Nachdem Streudiagramme schon im 19. Jahrhundert erfunden wurden [11], stellt sich natürlich die Frage, warum man sich noch mit einer so alten und elaborierten Technik befassen sollte. Doch William Cleveland et al. bemerkte schon in seiner Publikation:

> „Still, we can add graphical information to scatter plots to make them considerably more powerful.“ [8]

Clevelands Arbeit beinhaltete Verbesserungsvorschläge, welche einige Probleme, die das Anwenden der Streudiagrammtechnik mit sich bringen, lösen sollten. Schließlich mögen Streudiagramme noch so intuitiv und verständlich sein, sie kämpfen mit einigen methodologischen Problemen, welche den effektiven Einsatz im Bereich der visuellen Datenanalyse erschweren und ihren Nutzen schmälern. Auf einige dieser Probleme wird nun in den folgenden Absätzen etwas detaillierter eingegangen.

Typischerweise erstellt der menschliche Verstand beim Betrachten von Visualisierungen ein mentales Modell der Daten. Da Streudiagramme aber nur einzelne diskrete Punkte visualisieren, muss ein kontinuierliches Modell der Daten bewusst erarbeitet werden. Der menschliche Beobachter wird somit beim Interpretieren der Daten alleine gelassen und muss selber die Zusammenhänge zwischen den Dimensionen erschließen.

Zusätzlich sind die Daten, die meistens analysiert werden sollen, mehr als nur zweidimensional. Streudiagramme können aber per se nur zwei Dimensionen gleichzeitig darstellen, weil Einfärbung von Punkten nur schwer möglich ist. Um dieses Problem zu umgehen, wurden Streudiagramm – Matrizen eingeführt, welche für alle möglichen Paare von Dimensionen jeweils ein Streudiagramm beinhalten. Jedoch steigt die Anzahl der zu zeichnenden bzw. zu betrachtenden Streudiagramme quadratisch mit der Anzahl der Dimensionen. Allein für einen fünf dimensionalen Datensatz müssten schon 25 Streudiagramme exploriert werden. Diese fehlende Skalierbarkeit wird besonders deutlich, wenn man sich in Erinnerung ruft, dass im Bereich der Multimedia- oder Textanalyse schnell hundert- bis tausenddimensionale Datensätze entstehen können.

Die visuelle Datenanalyse mittels Streudiagrammen wäre deutlich einfacher und effizienter, wenn es möglich wäre, zumindest drei Dimensionen gleichzeitig zu visualisieren. Hierfür müssten die Punkte, wie schon oben erwähnt, eingefärbt werden, was nicht ohne Weiteres möglich ist. Schließlich können sich beispielsweise Punkte gegenseitig überdecken, und dadurch kann die Farbe der darunter liegenden Punkte nicht mehr wahrgenommen werden. Zudem erschwert die Überlagerung von Punkten die Abschätzung der Dichteverteilung.

Daten mit beliebigen Korrelationen zwischen den Dimensionen – seien sie global oder lokal – haben typischerweise die Eigenschaft, dass die Punkteverteilung bzw. die Dichte räumlich variiert. Häufig finden sich lokale Bereiche in den Daten, welche eine hohe Punktedichte haben. Diese Gebiete leiden unter einem geringen Detailreichtum, welcher lokale Effekte verschwinden lässt. Jedoch sind häufig gerade diese Bereiche für die Analyse und das Verständnis der Daten kritisch. Andererseits gibt es genauso häufig Gebiete, die wenige oder gar keine Punkte enthalten. Die Platzverschwendung durch Flächen mit niedriger Punktedichte führt dazu, dass Areale mit hoher Dichte keinen adäquaten Platz erhalten können.

Gerade der letzte Problempunkt soll an einem realen Beispiel weiter verdeutlicht werden, schließlich bieten sich bei zeitnahen Ereignissen Streudiagramme zur ersten Analyse an. Betrachtet man beispielsweise das lokale Phänomen Erdbeben, so kann jedes Erdbeben als Ereignis mit Längengrad, Breitengrad und Erdbebenstärke aufgefasst werden. Nichts liegt daher näher als einen solchen Datensatz mittels Streudiagrammen zu visualisieren. Jedoch wird man sehr schnell feststellen, dass Erdbeben bevorzugt lokal gehäuft auftreten und die genaue Analyse dadurch erheblich erschwert wird. Zur besseren Veranschaulichung wurde von der amerikanischen Behörde für geologische Beobachtungen (United States Geological Survey) ein Datensatz [22] verwendet. Dieser Datensatz beinhaltet alle weltweiten Erdbeben mit ei-

ner Stärke größer als 1.0 auf der Richterskala. Der beobachtete Zeitraum umfasst 7 Tage vom 26. Mai 2010 bis zum 2. Juni 2010, und insgesamt wurden 1027 Erdbeben verzeichnet. Um die lokale Häufung der Erdbeben sichtbar zu machen, wurde eine Dichteansicht auf die Daten generiert. Die verwendeten Farben stehen dabei für die Anzahl der Erdbeben (von rot bis hin zu gelb). Hierbei wurden zur besseren Orientierung die Ländergrenzen als Polygonzüge integriert. Das Ergebnis kann in Abbildung 1.1 begutachtet werden. Deutlich sichtbar sind einzelne Regionen, die besonders stark unter Erdbeben leiden, während andere (meist) völlig verschont bleiben. Und genau diese Ungleichverteilung macht die visuelle Analyse so schwierig, weil gerade die Bereiche ohne Erdbeben eigentlich gar nicht so interessant sind.

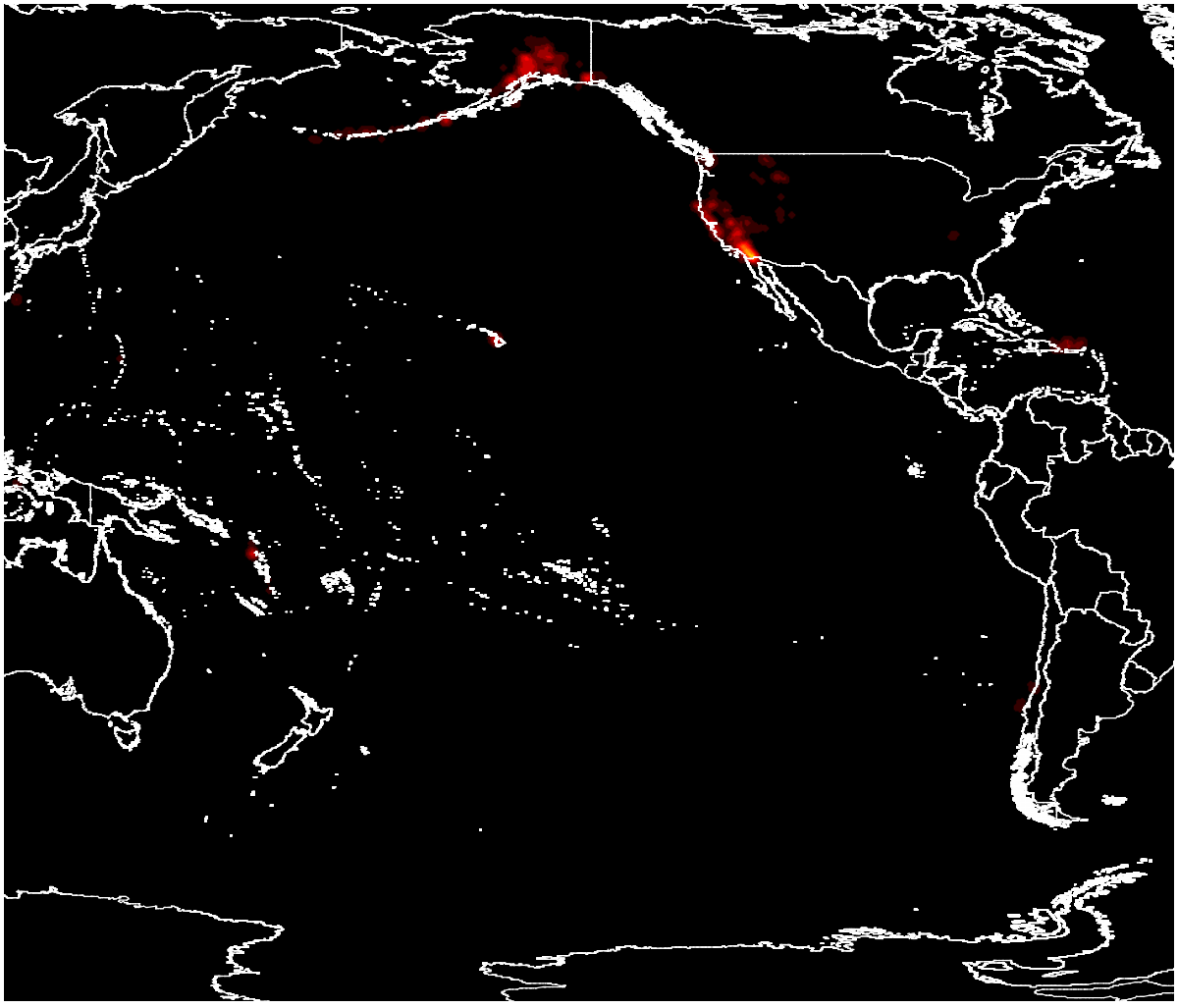


Abbildung 1.1: Dichtevisualisierung der weltweiten Erdbeben im Zeitraum vom 26. Mai 2010 bis zum 2. Juni 2010. Die verwendeten Farben repräsentieren die Anzahl an Erdbeben, wobei rot für eine niedrige Anzahl und gelb für eine hohe Anzahl steht. Die Ungleichverteilung der Erdbeben ist auf Grund der ungleichen Dichteverteilung deutlich sichtbar.

Gerade beim Explorieren großer Datensätze fällt der hohe Grad an Überdeckung von Punkten besonders stark negativ ins Gewicht. Dieser Hauptnachteil beim Visualisieren mit Streudiagrammen kann dazu führen, dass Korrelationen nicht sichtbar oder zumindest nur sehr schwer zu beobachten sind. Die vorliegende Arbeit befasst sich mit Lösungen für Punktüberdeckung und ungleiche Dichteverteilung. Hierbei werden nicht nur die verwendeten Techniken, sondern auch ein grundsätzliches Rahmenwerk vorgestellt, welches dem Benutzer starken Einfluss auf Verzerrungsgrad und Stärke des angewendeten Pixel Placements gewährt. Anders als bei schon existierenden Techniken, kann der Benutzer selber steuern, wie stark und mit welcher Technik er die Daten verzerren will. Zusätzlich ist der Benutzer frei in der Auswahl des Pixel Placement - Verfahrens und kann auch hier die Auswirkungen direkt beeinflussen. Selbstverständlich kann der Benutzer auch im verzerrten Datenraum Datenanalysetechniken anwenden, wie beispielsweise Clusteringtechniken oder Ähnliches.

Der Aufbau dieser Arbeit ist wie folgt: Im sich nun anschließenden Kapitel 2 werden die schon existierenden Techniken vorgestellt. Hierbei wird zwischen den Dichte visualisierenden und den verrzerrungsbasierten Verfahren unterschieden. Der neue Ansatz, bei dem der Datenanalyst den Grad an Überdeckung und Verzerrung steuern kann, wird in Kapitel 3 vorgestellt. Er ermöglicht, beliebige Sichten auf die Daten zu generieren, um verborgene Muster und Beziehungen in den Daten zu entdecken. Das Potenzial der Technik, kaum sichtbare Information erkennbar zu machen, wird in Kapitel 4 gezeigt, indem Verzerrung und Pixel Placement - Techniken auf reale Datensätze kombiniert angewendet werden. Die Anwendungsbeispiele werden zeigen, dass die Stärke der vorgestellten Technik zum einen auf der Kombination der beiden oben genannten Techniken und zum anderen auf dem interaktiven Einfluss des Benutzers auf die Stärke der Verzerrung bzw. des Pixel Placements beruht. Im Kapitel 5 werden bereits existierende Techniken mit dem hier vorgeschlagenen Ansatz verglichen, indem eine Evaluierung durchgeführt wird. Abgerundet wird die vorliegende Arbeit mit dem letzten Kapitel 6, in welchem eine Zusammenfassung und ein Ausblick auf mögliche zukünftige Forschungsarbeiten gegeben wird.

Diese Arbeit basiert hauptsächlich auf einem veröffentlichten Artikel im Information Visualization Journal. Die genaue Literaturangabe des Artikels mit dem Namen „Generalized scatter plots“ ist im Literaturverzeichnis unter [15] zu finden.

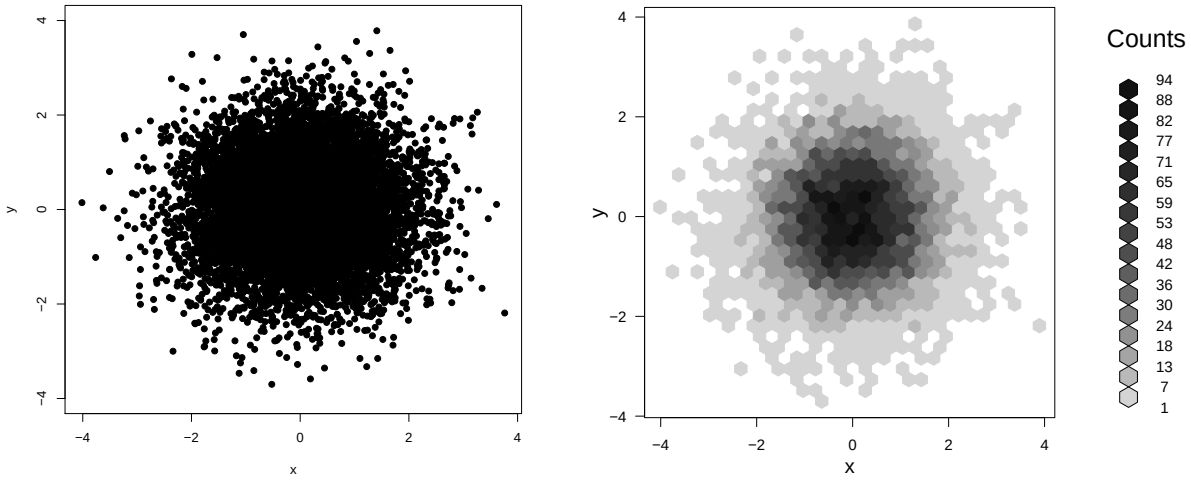
# 2 Existierende Ansätze

Schon in der Einleitung wurde erwähnt, dass Streudiagramme eine lange bekannte Visualisierungstechnik sind. Selbstverständlich sind die Probleme der Überdeckung und Ungleichverteilung der Daten schon früh erkannt worden. Daher gibt es viele Ansätze, welche die inhärenten Probleme der Streudiagramme lösen sollen. Im ersten Abschnitt dieses Kapitels sollen daher Techniken aufgeführt werden, welche das Überdeckungsproblem dadurch angehen, dass die Überdeckung auf das visuelle Attribut Helligkeit beziehungsweise Farbe gelegt wird. Dies sind also alles im weitesten Sinne Dichte visualisierende Verfahren, welche das Problem der Überdeckung nicht wirklich lösen. Als zweites werden die Ansätze vorgestellt, welche versuchen, mit einer irgendwie gearteten Verzerrung zumindest das Problem der Ungleichverteilung der Daten in den Griff zu bekommen. Abschließend sollen noch diejenigen Techniken, die nicht in die zwei vorherigen Abschnitte passen, wie beispielsweise stichprobenbasierte Ansätze, angeführt werden.

## 2.1 Dichtebasierte Visualisierungen

Mit dem Problem der Überdeckung hat sich beispielsweise der HexBin - Ansatz von Carr et al. [6] im Jahre 1987 befasst. Gerade zu dieser Zeit, die durch ein geringes Auflösungsvermögen der Ausgabegeräte geprägt war, waren große Datenmengen eine Herausforderung. Diese bestand zum einen in dem hohen Grad an Punktüberdeckung und zum anderen in der langen Erstellungsdauer einer Visualisierung. Carr ging beide Probleme an, indem die Anzahl der zu zeichnenden Objekte reduziert und die Dichte mittels Symbolen oder Farbskalen repräsentiert wurde. Zur Reduzierung der Datenmenge wurde der Datenraum mittels eines hexagonalen Gitters in Bereiche unterteilt. Die einzelnen Bereiche konnten nun mittels Symbolen repräsentiert oder einfach als Hexagone gezeichnet werden. Jedes der Hexagone wurde nur dann gezeichnet, wenn überhaupt ein Punkt enthalten war. Falls es aber gezeichnet wurde, konnte es passend zur Dichte eingefärbt werden. Der Hauptverwendungszweck dieser Technik lag darin, Streudiagramm - Matrizen besser zu visualisieren. Da die einzelnen Streudiagramme in solchen Matrizen typischerweise klein ausfallen, bietet eine Dichteansicht deutlich mehr Informationen als ein herkömmliches Streudiagramm. In Abbildung 2.1 wird zur Veranschaulichung ein Testdatensatz mit normalverteilten Zufallspunkten einmal als Streudiagramm (a) und einmal als HexBin - Grafik (b) gezeigt.

Zudem gab es auch die Idee, die Dichteinformation mittels gefüllter Konturlinien darzustellen. Diesem Gedankengang folgen die Vorschläge von Bowman in [3, 4]. Hierbei wird die Dichteverteilung des Eingabedatensatzes mit einer gegebenen Kernel - Funktion abschätzt. Für das in Abbildung 2.2 gezeigte Resultat dieser Technik wurde beispielsweise eine bivariate Normalverteilung zur Dichteabschätzung verwendet. Nach erfolgter Dichtebestimmung kann




(a) herkömmliches Streudiagramm (b) HexBin Visualisierung

Abbildung 2.1: Vergleich eines normalen Streudiagramms mit dem HexBin - Ansatz. Hierzu wurden 10 000 normalverteilte Zufallspunkte verwendet.

nun die eigentliche Visualisierung gezeichnet werden. Hierzu werden die Regionen, welche in einem bestimmten Dichtebereich liegen, mit der entsprechenden Farbe eingefärbt. Dieses Verfahren ist also auch eine Aggregation der Daten, jedoch detaillierter aufgelöst als beim oben vorgestellten HexBin - Ansatz.

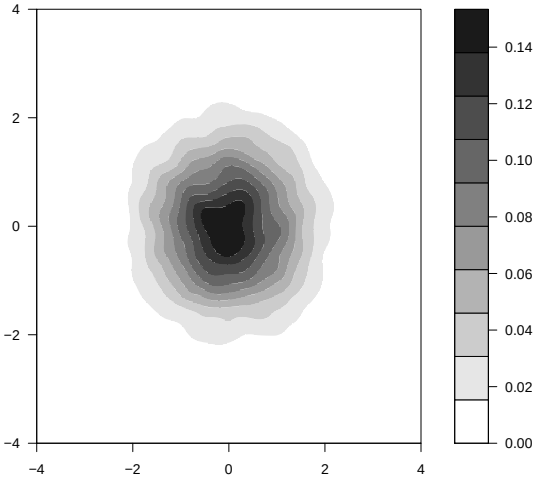



Abbildung 2.2: Visualisierung mittels gefüllter Konturlinien desselben Testdatensatzes.

Alternativ zu den oben vorgestellten aggregierenden Verfahren wird beispielsweise in einem Buch von Unwin [23] folgende Technik vorgeschlagen. Die Punkte werden genau wie beim normalen Streudiagramm gezeichnet mit nur einem Unterschied: jeder der Punkte ist semi-transparent. Dadurch bleibt eine große Ähnlichkeit zum traditionellen Streudiagramm erhalten, es wird jedoch zusätzlich die Dichteinformation visualisiert. Durch die Überzeichnung transparenter Datenpunkte werden dichtere Bereiche undurchsichtig und nicht so dichte

Bereiche bleiben durchscheinend. In Abbildung 2.3 wurde dieses Verfahren auf den schon mehrfach oben verwendeten Testdatensatz angewendet. Der große Nachteil dieses Ansatzes liegt darin, dass es kaum möglich ist, die Dichte abzuschätzen.

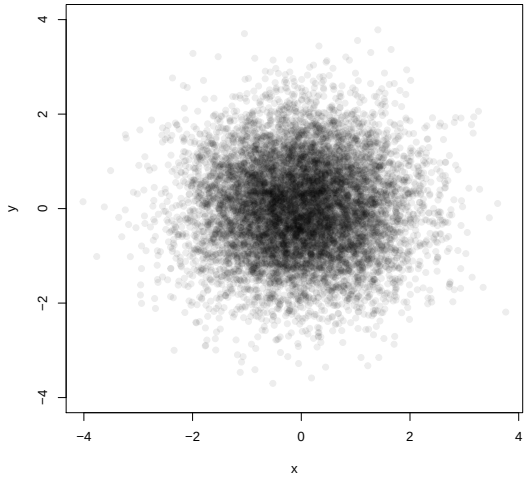


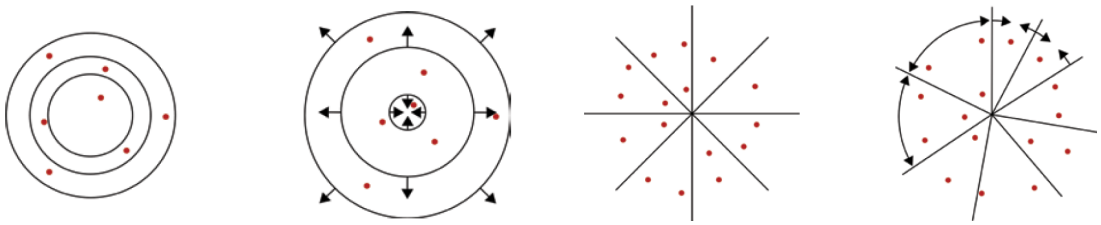

Abbildung 2.3: Visualisierung des Testdatensatzes mit semi-transparenten Punkten.

## 2.2 Verzerrungsbasierte Techniken

Eine einfache Lösung zur Vermeidung von Punktüberdeckung ist das Vergrößern der interessanten beziehungsweise dichten Regionen. Hierfür kann man entweder einfaches Zoomen [23] oder weiterführende Verfahren, wie FishEye - Lenses [14], verwenden. Diese Methoden bieten dem Benutzer einen höheren Detailreichtum in den vergrößerten Bereichen. Jedoch verliert man beispielsweise beim einfachen Zoomen den Überblick über den gesamten Datenraum. Zusätzlich ist die Bestimmung des optimalen Vergrößerungsfaktors eine Herausforderung, da jeder erhöhte Detailgrad mit einer geringeren Auflösung in nicht vergrößerten Bereichen gepaart ist.

Ähnlich zum vorhergehenden Verfahren werden auch bei den in [1] von Bak et al. vorgestellten Verzerrungstechniken wichtige Bereiche vergrößert und unwichtige verkleinert. Hierbei wurde angenommen, dass Bereiche mit hoher Dichte als wichtiger und interessanter gelten als Regionen mit geringer Dichte. In dieser Arbeit werden zwei verschiedene Ansätze zur Verzerrung des Datenraums vorgestellt. Zur besseren Veranschaulichung werden die schematischen Erklärungen in Abbildung 2.4 gezeigt. Beim MultiRadial - Verfahren (Abbildung 2.4(a)) werden zunächst die Dichtezentren bestimmt, und anschließend wird, ausgehend von diesen Dichtezentren, der Datenraum in konzentrische Gitterzellen unterteilt. Nun kann für jede der Gitterzellen die jeweilige Punktedichte bestimmt werden, und der konzentrische Ring entsprechend vergrößert beziehungsweise verkleinert werden. Die zweite vorgestellte Verzerrungstechnik nennt sich MultiAngular – schematisch dargestellt in Abbildung 2.4(b) – und bestimmt zuerst die Zentren der Daten, welche am wenigsten Dichte aufweisen. Anschließend wird der Datenraum ausgehend von den Zentren in strahlenförmige Bereiche unterteilt.

Für jeden dieser Bereiche wird nun wiederum die Dichte bestimmt, und die Regionen können gemäß der relativen Dichte verzerrt werden. Beide Verfahren wurden in die Generalized Scatter Plots integriert, um möglichst viele verschiedene Datenverteilungen unterstützen zu können.



(a) MultiRadial (b) MultiAngular

Abbildung 2.4: Schematische Darstellung der beiden von Bak et. al vorgestellten Verzerrungstechniken. Beide Grafiken wurden aus [1] übernommen.

In geographischen Anwendungsgebieten werden häufig Beobachtungen oder andere Attribute auf Karten dargestellt. Typischerweise entsprechen dabei die Kartenregionen nicht den visualisierten statistischen Größen. Die Technik, um die Regionen entsprechend zu verzerren, ist als Kartogramm bekannt und wurde in den Arbeiten von House und Kocmoud in [13] und von Keim et al. in [16] vorgestellt. Hierbei werden die Flächen der einzelnen Bereiche entsprechend den statistischen Werten vergrößert beziehungsweise verkleinert. Ziel dieser Verzerrung ist, dass die geographische Größe von Kartenregionen genau der statistischen Größe entspricht, wobei die Topologie möglichst erhalten bleiben soll. Ein Beispiel für die Kartogrammtechnik wird in Abbildung 2.5 gegeben. Diese Visualisierung zeigt gleichzeitig die Mehrheitsverhältnisse im jeweiligen Bundesstaat (Farbe) und die jeweilige Anzahl an Wahlmännern (Größe der Fläche).

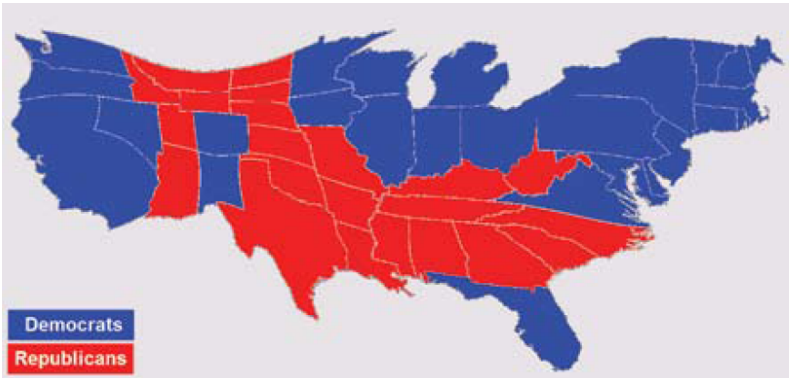



Abbildung 2.5: Kartogramm der Ergebnisse der amerikanischen Präsidentschaftswahl von 2008. Die Größe eines Staates entspricht der Anzahl an Wahlmännern und die Farbe gibt die jeweiligen Mehrheit im Staat an (rot für eine Mehrheit der Republikaner und blau für eine Mehrheit der Demokraten). Dieses Bild wurde aus [1] entnommen.

Eine weitere Technik zur überdeckungsfreien Darstellung großer Datenmengen wurde von Keim et al. in [19, 18] unter dem Namen PixelMaps vorgestellt. Der grundlegende Gedanke ist hierbei, durch eine Verzerrung des Datenraums Bereichen hoher Punktdichte mehr Platz zur Verfügung zu stellen, indem man nicht so dichte Bereiche verkleinert. Dies wird durch eine rekursive Aufteilung (ähnlich dem QuadTree - Verfahren) des Datenraums erreicht, wobei die einzelnen Gitterzellen anschließend gemäß der Dichte vergrößert beziehungsweise verkleinert werden. Zur Gewährleistung der Überdeckungsfreiheit werden erst die kleinsten Cluster eingezeichnet und erst anschließend die Punkte nächstgrößerer Cluster. Hierbei werden die Punkte an die nächstmögliche freie Stelle gesetzt, um Punktüberdeckungen zu verhindern. Ein Ergebnis dieser Technik, angewendet auf den in Kapitel 4.2 vorgestellten Datensatz, kann in Abbildung 2.6 begutachtet werden. Das Hauptproblem beim PixelMap - Verfahren liegt in der festen Verzerrung des Datenraums ohne Einflussnahme des Benutzers. Das Ergebnis dieser Technik ist also ein statisches Bild, bei dem der Benutzer den Grad an Überdeckung und Verzerrung nicht interaktiv steuern kann.

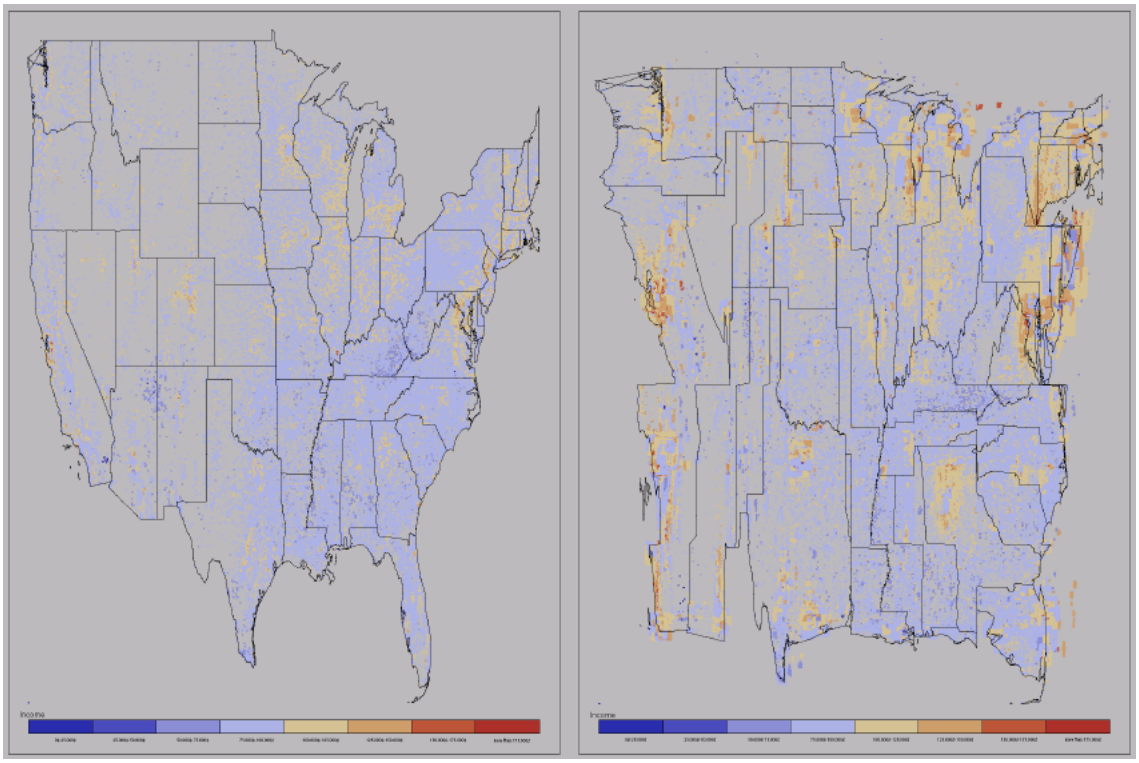


Abbildung 2.6: Ergebnis der PixelMap - Technik angewendet auf einen amerikanischen Zensusdatensatz zur Einkommensverteilung. Diese Grafik stammt aus der Arbeit von Keim et al. [18].

## 2.3 Weitere Ansätze

Zusätzlich zu den oben vorgestellten Techniken soll noch ein Verfahren angeführt werden, welches sich auch mit dem Problem der Überdeckung befasst. Hierbei handelt es sich um das Stichprobenziehen aus Daten, bei dem die geringere Datenmenge zu besseren visuellen Ergebnissen führen soll. Dieser Ansatz wurde beispielsweise unter dem Begriff *Sampling* in [9, 2] genau beschrieben. Es gibt dabei mehrere Varianten, so wird unter anderem unterschieden, ob man einfach eine zufällige Stichprobe zieht oder ob die Stichprobe repräsentativ ist. Für die Repräsentativität können dabei im Falle der Streudiagramme die räumliche Lage oder auch die Verteilung der Datenwerte der dritten Dimension berücksichtigt werden. Der große Vorteil dieser Technik ist, dass auf Grund der geringeren Datenmenge vorher verdeckte Muster sichtbar werden können. Zudem kann man mit einer guten Stichprobe trotz geringerer Datenmengen den gleichen visuellen Eindruck erzeugen. In Abbildung 2.7 wurde eine zehnprozentige, zufällige Stichprobe aus dem schon verwendeten und gezeigten Testdatensatz gezogen. Obwohl die Datenmenge deutlich, nämlich um eine Größenordnung, reduziert wurde, kann die zugrunde liegende Normalverteilung noch erkannt werden.

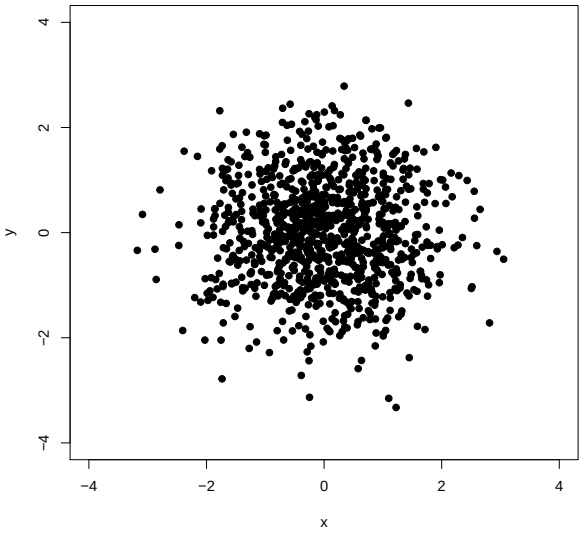
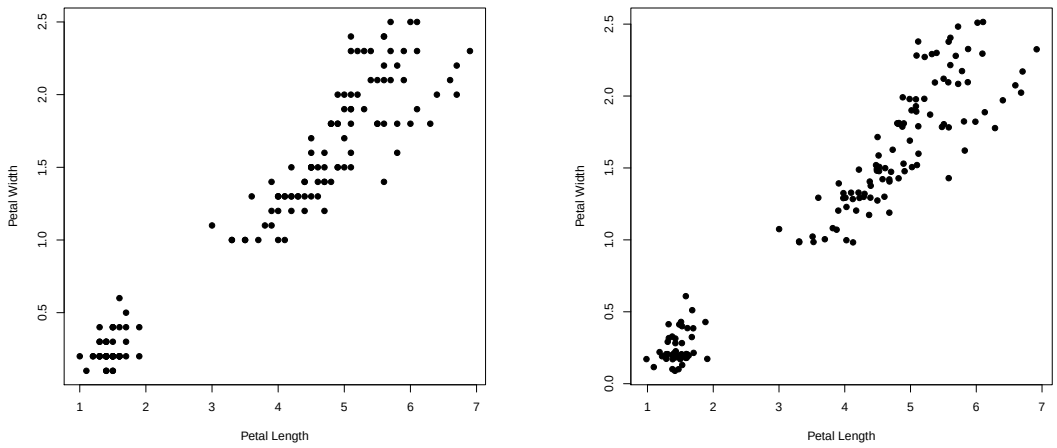



Abbildung 2.7: Visualisierung einer zehnprozentigen, zufälligen Stichprobe, gezogen aus dem Testdatensatz.

Ferner kann das Problem der Punktüberdeckung zumindest teilweise dadurch gelöst werden, dass man die Daten künstlich verrauscht. Im Buch von Chamber [7] wird dieses Vorgehen als *Jittering* bezeichnet und eingeführt. Hierbei werden alle Daten von ihrem Ursprungsort ein kleines Stück gemäß einer Normalverteilung wegbewegt. Durch die zufällige Verteilung der Punkte können bei geringen Punktedichten tatsächlich Erfolge erzielt werden. Allerdings versagt dieser Ansatz bei hohen Graden an Punktüberdeckungen, weil ein geringes Verrauschen der Daten nicht mehr ausreicht. Zudem werden auch Punkte versetzt, bei denen gar keine Notwendigkeit vorliegt, dies zu tun, weil keine Überdeckung vorherrscht. In Abbildung 2.8 werden als Beispiel zwei Dimensionen des Iris - Datensatzes, nämlich „Petal Length“ (Blütenblattlänge) und „Petal Width“ (Blütenblattbreite), gegeneinander abgetragen. Dabei wurde

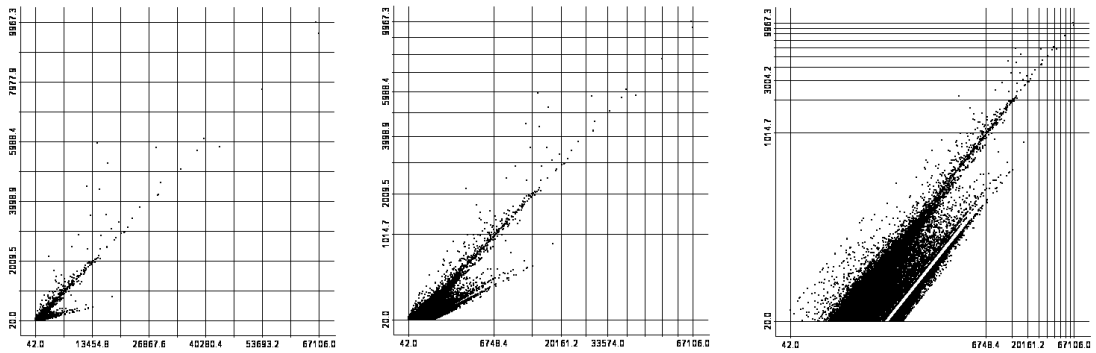
in der linken Grafik auf künstliches Verrauschen der Daten verzichtet, während in der rechten Grafik diese Technik zur Bekämpfung der Punktüberdeckung eingesetzt wurde. Es ist hierbei deutlich ersichtlich, dass sich im Ursprungsdatensatz Punkte überdecken. Dies kann durch ein leichtes Verrauschen der Daten behoben werden, wobei auch sich nicht überdeckende Punkte von ihrer Ursprungsposition versetzt werden.




(a) kein künstliches Rauschen (b) mit künstlichem Rauschen

Abbildung 2.8: Streudiagramm zweier Dimensionen des Iris - Datensatzes zur Veranschaulichung der Jittering - Technik. Auf der x-Achse ist die Blütenblattlänge und auf der y-Achse ist die Blütenblattbreite abgetragen.

Außerdem ist bei der explorativen Datenanalyse in der Statistik auch eine Änderung der Achsenskalierung üblich. Anders als bei der gebräuchlichen linearen Skalierung der Achsen werden dabei logarithmische oder auch Quadratwurzel- Skalierungen verwendet. Dies dient dazu, eine der Datenverteilung angepasste Visualisierung zu erreichen. Als Beispiel hierfür wurde auf der nächsten Seite in Abbildung 2.9 der Telefondatensatz aus dem Kapitel 4 mit verschiedenen Achsenskalierungen visualisiert. Für den hier abgebildeten Datensatz scheint eine logarithmische Skalierung der Achsen am besten zu sein.




(a) Lineare Skalierung (b) Quadratwurzel-Skalierung (c) Logarithmische Skalierung

Abbildung 2.9: Auswirkungen unterschiedlicher Achsenskalierungen auf den Telefondatensatz aus Kapitel 4.

# 3 Generalized Scatter Plots

Eines der Hauptprobleme bei der Visualisierung mittels Streudiagrammen ist auf eine typische Eigenschaft der Daten zurückzuführen: üblicherweise besitzen Datensätze eine inhärente Dichteungleichverteilung. Schließlich besteht meist eine irgendwie geartete Relation zwischen den visualisierten Dimensionen, welche sich in globalen und / oder lokalen Korrelationen äußert. Diese Korrelationen führen nun zu Bereichen mit hoher Punktedichte und Bereichen mit niedriger Punktedichte.

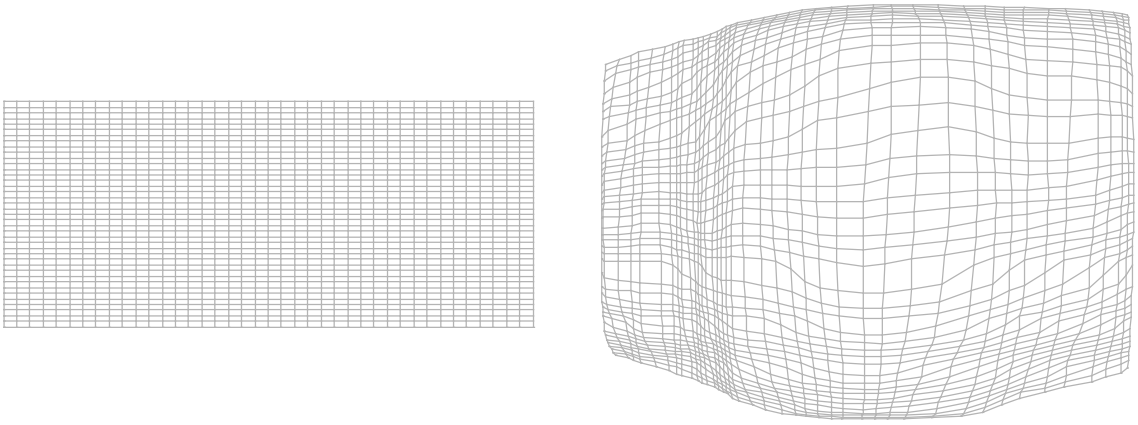
An und für sich ist dieser Vorgang genau das, was man eigentlich erreichen und beobachten will. Jedoch resultiert hieraus auch ein entscheidendes Hindernis für die visuelle Datenanalyse. Sollte die Punktedichte nämlich zu groß sein, reicht das Auflösungsvermögen des Bildschirms nicht mehr aus, um alle Punkte überdeckungsfrei darzustellen. Dadurch kann zum einen das Auffinden lokaler Trends und zum anderen das Abschätzen der Punktedichte unmöglich werden.

Somit wird eine Verzerrung des Datenraums benötigt, welche lokale Korrelationen auch bei hoher Punktedichte wieder sichtbar macht. Dabei muss aber auch der globale Zusammenhang zwischen den Dimensionen weiterhin richtig dargestellt werden. Die Grundidee all der hier vorgestellten Verfahren ist, Bereichen mit hoher Punktedichte mehr Platz zu gewähren und Bereiche mit niedriger Punktedichte räumlich zu komprimieren. Das Ergebnis sollte somit eine Gleichverteilung der Punktedichte im gesamten Datenraum sein. Durch die Gleichverteilung werden lokale Korrelationen dichter Regionen ersichtlich und der zu Verfügung stehende Raum wird besser ausgenutzt. Schließlich werden beispielsweise leere Bereiche nach der Verzerrung sehr stark geschrumpft sein, um Platz zu schaffen.

Neben der eigentlichen Verzerrung der Daten rückt hierbei der Benutzungsaspekt in den Mittelpunkt der Aufmerksamkeit. Wird der Datenraum einfach nur – sozusagen ohne Vorwarnung – vollständig verzerrt, so kann das mentale Modell des Benutzers und Betrachters nicht folgen. Nur wenn der Benutzer den Verzerrungsvorgang versteht und interaktiv verfolgen kann, so ist er in der Lage, trotz Verzerrung des Datenraums eine effektive Datenanalyse durchzuführen. Das Hauptaugenmerk bei den Verzerrungstechniken muss also auf der interaktiven Einflussnahme des Benutzers auf den Verzerrungsprozess liegen. Dabei sollte der Benutzer jede mögliche Zwischenstufe zwischen der originalen und der vollständig verzerrten Ansicht generieren können.

Jedoch birgt das Verzerren des Datenraums immer auch Gefahren, da die visuelle Analyse erschwert wird. So müssen Beobachtungen im verzerrten Raum erst auf den originalen Datenzustand übertragen werden. Da dieser Vorgang nicht ohne bewusste mentale Anstrengung durchgeführt werden kann, sollte der Benutzer so gut wie nur möglich dabei unterstützt werden. Der Benutzer soll seine volle Aufmerksamkeit den Daten widmen können und nicht über die Auswirkungen der Verzerrung nachdenken müssen. Eine Möglichkeit zur Entlastung des Benutzers ist die Einzeichnung von regulären Gitterzellen. Diese Gitterzellen werden zusam-

men mit den Datenpunkten gleichzeitig verzerrt. Anhand der Verzerrung des Gitters können Rückschlüsse auf die Wirkungsweise der Verzerrungen erzielt werden. In Abbildung 3.1 wird zur Veranschaulichung ein unverzerrtes und ein verzerrtes Gitter gezeigt.



(a) ursprünglicher Datenraum (b) verzerrter Datenraum

Abbildung 3.1: Diese Abbildung zeigt exemplarisch, wie das Gitternetz durch die angewendeten Verzerrungstechniken synchron zum Datenraum verzerrt wird. Im verzerrten Zustand fördert es das Verständnis der eingesetzten Verzerrungen zur effektiveren und effizienteren Datenanalyse. Für die vorliegende Abbildung wurden zwei verschiedene Verzerrungen (MultiRadial und HistoScale) miteinander kombiniert.

Um Abbildung 3.1 zu erzeugen, wurde ein Zensusdatensatz der Vereinigten Staaten von Amerika über das Haushaltseinkommen verwendet. Hierbei steht jeder Dateneintrag für mehrere Haushalte, was zu einer deutlichen, geographischen Ungleichverteilung der Datenpunkte führt. Schließlich wohnen beispielsweise in den Wüstenstaaten nicht so viele Menschen wie an der bevölkerungsreichen Ostküste. In Abbildung 3.1(b) sind die Wüstenstaaten sichtbar zusammengeschnürt worden, während andere Staaten gleich groß geblieben oder noch größer geworden sind.

Durch das Verwenden von Gitternetzlinien können die Auswirkungen der angewendeten Verzerrungen einfach verständlich gemacht werden. Hierbei wird der Vergleich zum unverzerrten Datenraum ermöglicht, und die visuelle Datenanalyse kann trotz Verzerrungen effektiv und effizient durchgeführt werden.

Nachdem im weiteren Verlauf gezeigt wird, wie eine Ungleichverteilung der Daten mittels Verzerrungstechniken gemindert werden kann, soll der darauf folgende Abschnitt zeigen, wie man Überdeckung von Datenpunkten vollständig verhindert. Schließlich kann im Allgemeinen nach Verzerrung des Datenraums nicht ausgeschlossen werden, dass sich noch immer Punkte gegenseitig überdecken. Für diese Überdeckung von Punkten im Bildraum kann es genau zwei Möglichkeiten geben. Zum einen können die Datenpunkte im Datenraum genau gleiche Koordinaten besitzen und daher auch nach der Projektion in den Bildraum auf das gleiche Pixel abgebildet werden. Zum anderen ist es auch möglich, dass die Punkte im Daten-

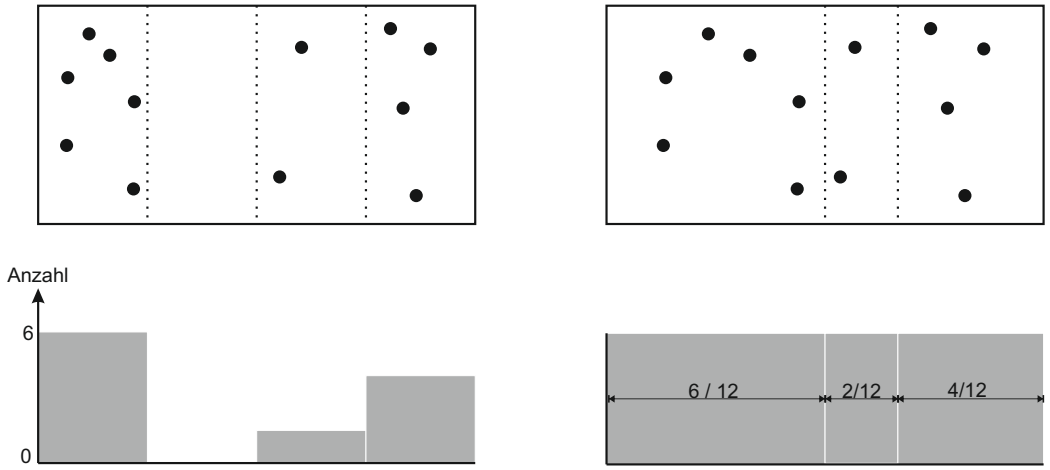
raum unterschiedliche Koordinaten haben, aber nach der Projektion auf dasselbe Pixel fallen. Dies ist typischerweise Folge eines zu geringen Auflösungsvermögens des Bildraums.

Aus dieser Beobachtung folgt, dass nur ein Zusammenspiel von Verzerrung und Pixel versetzenden Techniken in der in der Lage ist, gute Visualisierungen zu erzeugen. Genau diesem Ansatz folgen die Generalized Scatter Plots: der Benutzer hat direkten Einfluss auf Verzerrungsverfahren und Überdeckungsgrad. Ferner unterstützen sie den Benutzer bei der visuellen Datenanalyse mit Analysetools, wie Clustering mit Voronoizelleneinzeichnung, und mit dynamisch generierten Dichteansichten. Beginnen wird dieses Kapitel mit einer Einführung in das hauptsächlich verwendete Verzerrungsverfahren, anschließend werden verschiedene Pixel Placement - Techniken vorgestellt. Danach wird gezeigt, dass die Generalized Scatter Plots dem Benutzer helfen, ein Optimierungsproblem zu lösen, und abschließend wird die Referenzimplementierung kurz angerissen.

## 3.1 HistoScale

Das HistoScale - Verfahren wurde in der Arbeit von Keim et al. in [17] vorgeschlagen und bietet eine sehr einfache achsenparallele Verzerrung des Datenraums. Wie der Name schon sagt, wird die Dichteverteilung zunächst mittels eines Histogramms abgeschätzt. Auf Grund der so gewonnenen Verteilungsinformation kann anschließend der Datenraum verzerrt werden. Hierbei werden Bereiche mit hoher Dichte vergrößert und Bereiche mit niedriger Dichte verkleinert. Oder anders ausgedrückt: die Bereiche werden entsprechend ihren relativen Punktedichten skaliert. Die Skalierung ist dabei eine Streckung bzw. Stauchung der entsprechenden Achsenabschnitte und führt daher zu einer achsenparallelen Verzerrung. Vorteil dieses Verfahrens ist eine hohe Verständlichkeit und die Möglichkeit zur intuitiven Interpretation des Ergebnisses. Schwerwiegender Nachteil ist jedoch die reine achsenparallele Verzerrung des Datenraums. Schließlich ist nicht garantiert, dass die Datenpunkte immer passend liegen. Beispielsweise würde eine X-förmige Datenverteilung vom reinen HistoScale - Verfahren – bei entsprechenden Dichten – erst gar nicht verzerrt werden. In den folgenden Abschnitten wird detaillierter auf die Technik eingegangen. Zur besseren Verständlichkeit wird die Vorgehensweise in Abbildung 3.2 für die Verzerrung einer Dimension schematisch dargestellt.

Wie schon oben beschrieben, wird die Dichteverteilung ähnlich einer Histogrammberechnung ermittelt. Die Dichte wird dabei für jede Dimension einzeln erfasst und jede Dimension wird für sich alleine genommen verzerrt. Der Datenraum wird also zunächst in einer Dimension gleichmäßig in Intervalle unterteilt und anschließend wird in jedem Bereich die jeweilige Punktedichte ermittelt. Das schematische Vorgehen wird anhand eines Beispiels verdeutlicht und in Abbildung 3.2(a) dargestellt. Im vorliegenden Beispiel wurde der gesamte Datenraum entlang der x-Achse in vier Abschnitte aufgeteilt. In jedem wird nun die Anzahl der enthaltenden Datenpunkte berechnet und in einer Datenstruktur abgelegt. Im nun folgenden Verzerrungsschritt werden die Intervalle gemäß der jeweiligen Punktdichte verbreitert beziehungsweise geschmälert. Zur Bestimmung der neuen Position eines Datenpunktes müssen dabei zwei Berechnungen durchgeführt werden. Zum einen müssen die neue Breiten aller links vom Datenpunkt liegenden aufsummiert werden. Und zum anderen muss anhand der relativen Position im beinhaltenden Intervall die neue Position entsprechend der Verzerrung berechnet werden.




(a) Bestimmung der Punktanzahl für jedes Intervall im unverzerrten Originalzustand

(b) verzerrter Datenraum gemäß relativer Punktedichte

Abbildung 3.2: Schematisches Vorgehen beim Verzerren des Datenraums mittels der HistoScale - Technik. Auf der linken Seite ist der ursprüngliche Datenraum abgebildet, in dem die Dichteverteilung in gleich breiten Bereichen bestimmt wird. Auf der rechten Seite ist das Endergebnis der Verzerrung zu sehen. Im unteren Bereich des Bildes wird verdeutlicht, dass die Verzerrung der Umwandlung eines Histogramms mit gleich breiten Säulen in ein Histogramm mit gleich hohen Säulen entspricht.

Zur Bestimmung der Skalierung der Intervallbreite wird die zuvor bestimmte relative Punktdichte verwendet. Hierbei ist die zu Grunde liegende Idee, dass jede Intervallbreite genau der relativen Punktdichte entsprechen soll. Dieser Überlegung folgt die folgende Formel 3.1 zur Berechnung der skalierten Breite eines Intervalls. Dafür muss nur das Verhältnis der Punktanzahl im Intervall $i$ zu der Anzahl aller Datenpunkte berechnet werden.

$$binWidth(i) = \frac{noOfPointsInBin(i)}{noOfAllPoints} \tag{3.1}$$

Für die Position eines Punktes im beinhaltenden Intervall darf nun aber nicht die gesamte Breite berücksichtigt werden. Es muss hierfür nämlich erst die lokale, relative Position im Intervall bestimmt werden und diese wird mit der in Formel 3.1 berechneten Breite multipliziert werden.

Somit ergibt sich für die Verzerrung eines Datenpunktes $p$ – im Intervall $int$ liegend – entlang der x-Achse folgende Berechnungsvorschrift 3.2:

$$distortedPosX(p,int) = \sum_{\substack{\text{abgeschlossene} \\ \text{Intervalle } i \\ \text{links von } p}} binWidth(i) + \frac{p.x - int.start}{int.end - int.start} \cdot binWidth(int) \tag{3.2}$$

Derselbe oben beschriebene Vorgang, sowohl Bestimmung der Punktedichte als auch Reskalierung der Intervalle, wird natürlich auch für die zweite Dimension wiederholt. Beide

Vorgänge können hierbei absolut unabhängig voneinander durchgeführt werden und bieten somit die Möglichkeit zur Parallelisierung.

Ebenso gut könnte auch ein Gitter gemäß der α-Quantile erzeugt werden, um die jeweilige Datendichte zu bestimmen. Dabei würde im nächsten Schritt das Gitter zu einem regelmäßigen Gitter verzerrt werden. Dieser Ansatz würde etwas bessere Ergebnisse liefern, da gerade Bereiche mit hoher Punktedichte mit mehr Gitterzellen versehen würden. Jedoch wurde in der Referenzimplementierung auf Grund der einfacheren Umsetzung der erste Ansatz verfolgt. Zusätzlich birgt der erste Ansatz Laufzeitvorteile, da hierbei die Höhe und Breite einer jeden Gitterzelle gleich ist und dies einige Berechnungen beschleunigt.

Wie schon eingangs erwähnt, bietet die rein achsenparallele Verzerrung nicht nur Vorteile sondern auch Nachteile. Schließlich halten sich reale Daten meist nicht an Achsenparallelismus. Daher zeigt der folgende Absatz nun einen eigenen Lösungsweg auf, wie man weiterhin das oben beschriebene Verfahren verwendet und trotzdem eine der Datenverteilung gerechte Verzerrung erreicht. In Abbildung 3.3 wird das schematische Vorgehen an Hand eines kleinen Beispieldatensatzes verdeutlicht.

In einem Vorverarbeitungsschritt lässt man den Eingangsdatenraum aus Abbildung 3.3(a) jeweils in fünf Grad - Schritten rotieren. Nach jedem Rotationsschritt fasst man den so transformierten Raum als zu verzerrenden auf und berechnet die Dichteverteilung gemäß der oben beschriebenen Intervalltechnik. Anschließend wird die Entropie der Dichteverteilungen für die beiden Projektionen auf die neuen Hauptachsen berechnet. Die beiden Entropiewerte werden danach miteinander kombiniert, beispielsweise durch Summenbildung. Diese Entropiewerte werden nun für alle 72 Rotationschritte berechnet und hinterher kann die Transformation bestimmt werden, welche in der maximalen Entropie der Dichteverteilungen resultiert. In dem auf diese Art und Weise bestimmten und rotierten Datenraum in Abbildung 3.3(b) wird nun die bereits vorgestellte HistoScale - Technik zur Verzerrung angewendet. Schließlich kann das verzerrte Ergebnis aus Abbildung 3.3(c) in den ursprünglichen Raum zurück rotiert werden, was in Abbildung 3.3(d) zu sehen ist. Auch wenn HistoScale im transformierten Raum nur eine achsenparallele Verzerrung durchführt, so kann durch die Rotation eine Verzerrung entlang quasi beliebiger Basisvektoren erfolgen.

Da der Fokus dieser Arbeit insbesondere auf der interaktiven Einflussnahme des Benutzers auf die Datentransformationen lag, muss eine Möglichkeit zur Steuerung der Verzerrungstärke gegeben sein. Die für den Benutzer einfach verständliche Lösung wäre ein Schieberegler, welcher eine stufenlose Transformation des ursprünglichen Zustands in den verzerrten Endzustand ermöglicht. Dabei wäre dann beispielsweise am linken Ende des Reglers der ursprüngliche Zustand zu finden und je weiter der Schieberegler nach rechts geschoben wird, desto stärker wird der Datenraum verzerrt. Implementiertungstechnisch gesehen, wäre es möglich, dies beispielsweise mittels linearer Interpolation umzusetzen. Die einfache Berechnungsvorschrift zur Bestimmung jeglicher Zwischenformen kann in Formel 3.3 betrachtet werden.

$$\overrightarrow{paintPos} = (1-\alpha) \cdot \overrightarrow{origPos} + \alpha \cdot \overrightarrow{distortedPos} \tag{3.3}$$

Der skalare Parameter α spiegelt hierbei die Stellung des Schiebereglers wieder und liegt im Intervall von $[0,1]$. Durch ihn wird die beliebig gewichtete lineare Interpolation zwischen ursprünglicher Position und verzerrter Position ermöglicht.

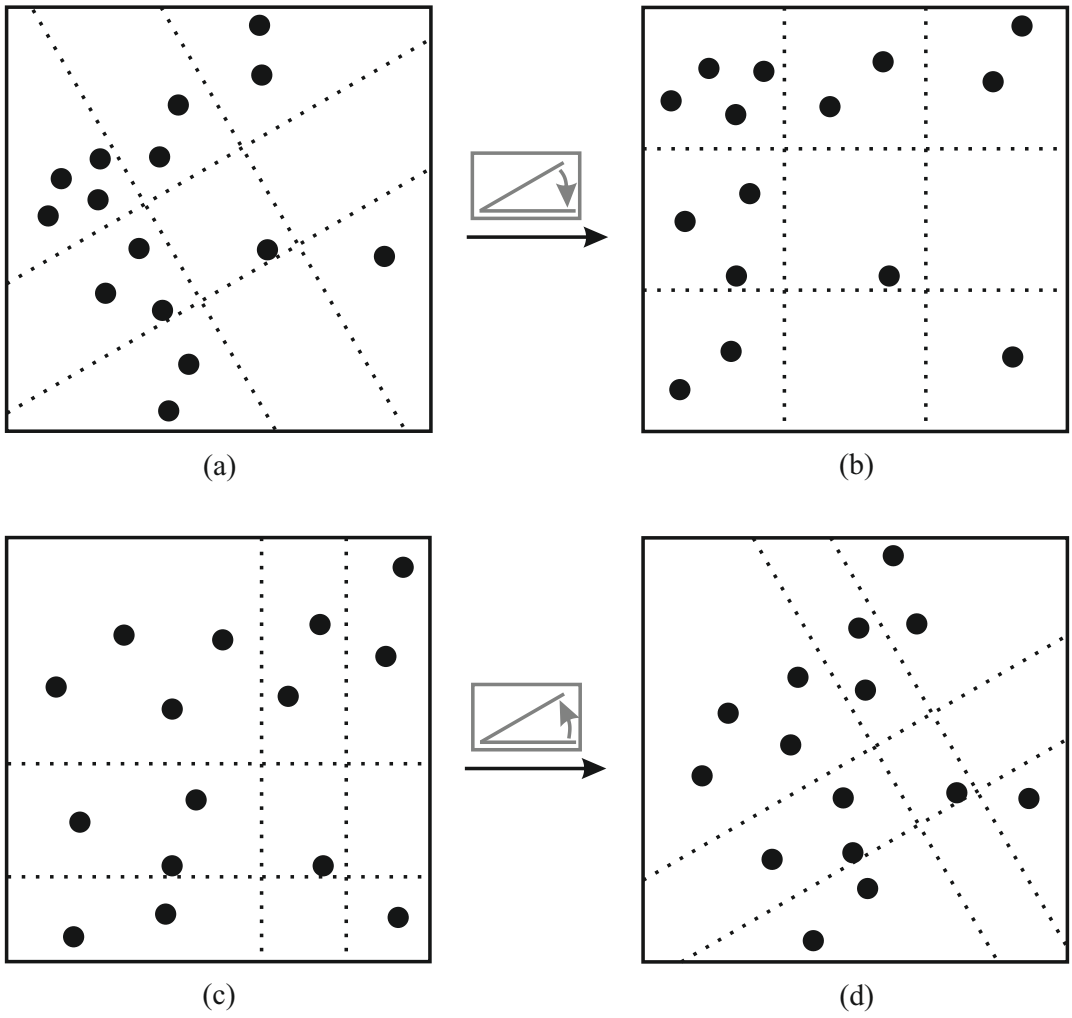



Abbildung 3.3: Diese Graphiken zeigen das grundsätzliche Vorgehen zur Erweiterung des HistoScale - Ansatzes. In (a) ist der Eingangsdatensatz zu sehen, dieser wird optimal rotiert, was in (b) resultiert. Der so transformierte Raum wird als Eingabe für das oben vorgestellte HistoScale - Verfahren verwendet und der transformierte Datenraum wird – in (c) sichtbar – verzerrt. Abschließend wird in (d) der transformierte und verzerrte Raum in die Originalposition zurück gedreht.

## 3.2 Pixel Placement

Wie schon oben erwähnt, reicht reine Verzerrung meistens nicht aus, um Überdeckungsfreiheit zu garantieren. Die Pixel Placement - Technik nimmt nun direkten Einfluss auf die Koordinaten eines Punktes im Bildraum, um Überdeckung von Punkten zu verhindern. Die grundsätzliche Idee des Verfahrens ist, Datenpunkte an ihre Originalposition zu platzieren, wenn sie frei ist bzw. ihnen sonst die nächstmögliche freie Stelle zuzuweisen. Hierbei ist die Reihenfolge der Platzierung sehr wichtig und darf nicht vernachlässigt werden, da sonst visuelle Artefakte

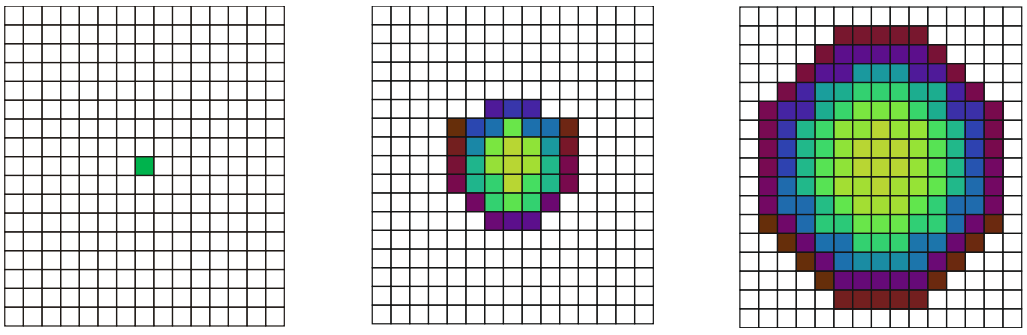
entstehen können. Für beide nun folgende Verfahren wird angenommen, dass sich die Punkte bereits in einer sinnvollen Reihenfolge befinden. Der implementierte Prototyp sortiert die Datenpunkte nach der dritten Dimension, welche mittels Farbe visualisiert wird. Hierbei kann der Benutzer auswählen, ob aufsteigend oder absteigend sortiert werden soll.

Im Folgenden werden zwei verschiedene Pixel Placement - Verfahren vorgestellt. Das erste Verfahren garantiert gegebenenfalls, dass jedes Pixel nur maximal einen Datenpunkt repräsentiert. Jedoch ist dieser Ansatz rechenintensiver und daher wird im anschließenden Abschnitt ein heuristischer und nicht so laufzeitintensiver Algorithmus vorgestellt, welcher jedoch nicht mehr ausschließen kann, dass sich Punkte überdecken.

### 3.2.1 Exhaustive Pixel Placement

In diesem Abschnitt soll eine Technik vorgestellt werden, welche dafür sorgt, dass jedes Pixel nur maximal $n$ Datenpunkte repräsentiert. Der Benutzer hat dabei direkten Einfluss auf den Wert $n$, indem er angibt, wie viel Prozent der ursprünglichen Überdeckung er zulassen will. Beispielsweise könnte er nur zehn Prozent Überdeckung zulassen, was bedeuten würde, dass an einer Pixelposition, auf der vorher 100 Datenpunkte lagen, nach dem Pixel Placement nur noch 10 Datenpunkte liegen. Die restlichen 90 Datenpunkte werden dann in der Umgebung der ursprünglichen Position angeordnet. Eine detaillierte Beschreibung der Auswirkungen und der Umsetzung des Parameters wird im weiteren Verlauf vorgenommen.

Das grundsätzliche Vorgehen des hier vorgestellten Algorithmus lautet wie folgt: Die geordneten Datenpunkte werden einzeln abgearbeitet und jedem wird – je nach Anzahl der Punkte, die schon an derselben Position platziert sind – entweder die Originalposition oder die nächstmögliche Stelle zugewiesen. Die Auswirkungen des Exhaustive Pixel Placement auf einen Testdatensatz mit 155 Punkten mit denselben Koordinaten wird in Abbildung 3.4 gezeigt.



(a) Kein Pixel Placement angewendet (100 % der ursprünglichen Überdeckung) – 155 Punkte überdecken sich

(b) 50 % des maximalen Pixel Placements – auf jedem Pixel liegen maximal 4 Datenpunkte

(c) Keinerlei Überdeckung mehr – jedes Pixel enthält maximal einen Datenpunkt

Abbildung 3.4: Diese Abbildung zeigt, wie der Exhaustive Pixel Placement - Algorithmus auf einen Testdatensatz von 155 Punkten angewendet wird. Die einzelnen Graphiken zeigen dabei den Originaldatensatz (links) und zwei Ergebnisse bei unterschiedlichen Einstellungen für die Anzahl der erlaubten Überdeckungen.

In den anschließenden Abschnitten wird nun näher auf das Verfahren eingegangen. Die Hauptmethode der Exhaustive Pixel Placement -Technik wird im folgenden Algorithmus vorgestellt:

**Algorithmus 1:** Hauptmethode des Exhaustive Pixel Placement - Algorithmus

**Daten**: Geordnete Liste von Datenobjekten
**Ergebnis**: Neu berechnete Zeichenpositionen der Datenobjekte

```
1 int[ ][ ] overlapCount := new int[width][height];
2 foreach Objekt o aus der Lister aller Datenobjekte do
3     Point p := o.getPixelCoord();
4     if overlapCount[p.x][p.y] < maxAllowedOverlap then
5         o.setPaintCoord(p);
6         overlapCount[p.x][p.y]++;
7     else
8         rearrangeDataObject(o, p, overlapCount);
```

Die Hauptmethode des Pixel Placement - Verfahrens – im Algorithmus 1 dargelegt – besteht aus nachstehenden Schritten. Zunächst wird in Zeile 1 für den gesamten Bildraum ein Puffer angelegt, welcher für jedes Pixel speichert, wie viele Datenpunkte an dem jeweiligen Pixel schon platziert sind. Anschließend wird die Liste aller Datenpunkte in der gegebenen Reihenfolge durchlaufen. In Zeile 4 wird für jeden Datenpunkt anschließend überprüft, ob seine Originalposition im Bildraum noch zur Disposition steht. Falls dies der Fall ist, so kann der Datenpunkt an seiner Originalposition bleiben. Ferner wird im Puffer vermerkt, dass an den spezifischen Pixelkoordinaten nun ein zusätzlicher Datenpunkt liegt. Im gegenteiligen Fall jedoch muss der Punkt zu anderen Koordinaten verschoben werden. Die Berechnung neuer Koordinaten erfolgt in der Methode *rearrangeDataObject*, welche in Algorithmus 2 vorgestellt wird.

Die Methode *rearrangeDataObject* versetzt einen einzelnen Datenpunkt zur nächstmöglichen Stelle. Hierbei werden die Pixel kreisförmig um die Originalposition herum verschoben, jedoch wären beliebige andere Formen, wie beispielsweise quadratische oder elliptische Formen, möglich und auch einfach implementierbar. Um die nächstmögliche Stelle zu bestimmen, werden alle Pixel einer Kreislinie mit der Dicke von zwei Pixeln berechnet. Dabei wird der Radius der Kreislinie solange inkrementell erhöht, bis eine mögliche Pixelposition gefunden wurde. Zur Beschleunigung des Verfahrens wird für jedes Pixel im Bildraum gespeichert, welcher Radius zuletzt verwendet wurde. Daher wird in Zeile 1 angefragt, welchen Radius der zuletzt verwendete Kreis um die gegebenen Originalkoordinaten hatte. Zu Beginn des gesamten Pixel Placement - Verfahrens wurde dieser Wert für alle Pixel mit eins initialisiert. Der so gewonnene Radius kann nun dazu verwendet werden, mittels der weiter unten gezeigten Methode *calcNewPositions* sämtliche Pixel auf der Kreislinie um den gegebenen Mittelpunkt zu berechnen. Anschließend werden in Zeile 6 alle Koordinaten daraufhin überprüft, ob die Anzahl der schon platzierten Punkte die Anzahl der maximal erlaubten Punkte übersteigt. Falls sämtliche Koordinaten der Kreislinie abgearbeitet und keine neue Position gefunden wurde,

**Algorithmus 2:** Methode *rearrangeDataObject*
Versetzt einzelne Datenobjekte an die nächstmögliche Position

**Eingabe**: Zu verschiebendes Datenobjekt *o*, Originalpixelposition *p*, Array mit den Anzahl Objekten pro Pixel *overlapCount*
**Ergebnis**: Dem übergebenen Datenobjekt wurde eine neue Position zugewiesen

```
1  int radius := getLastUsedRadius(p);
2  Point [ ] newPositions := calcNewPositions(p, radius);
3  while neue Position noch nicht gefunden do
4      if newPositions beinhaltet noch nicht überprüfte Position then
5          Point nextPoint := nächster Punkt aus newPositions;
6          if overlapCount[nextPoint.x][nextPoint.y] < maxAllowedOverlap then
7              o.setPaintCoord(nextPoint);
8              overlapCount[nextPoint.x][nextPoint.y]++;
9      else
10         radius++;
11         newPositions := calcNewPositions(p, radius);
12 updateLastUsedRadius(p, radius);
```

werden in den Zeilen 10 und 11 alle Pixel einer Kreislinie mit einem um eins erhöhten Radius berechnet. Nun kann die Untersuchung der berechneten Koordinaten erneut durchgeführt werden. Dieser Vorgang wird solange wiederholt, bis eine passende Position gefunden wurde. Abschließend wird in Zeile 12 – wie schon oben beschrieben – der zuletzt verwendete Radius zur Verringerung der Laufzeit gespeichert.

Besonders kritisch für die Laufzeit des hier vorgestellten Verfahrens ist die Berechnung der Kreislinie. Schließlich wird für jedes Pixel, das versetzt werden muss, ein bzw. mehrere Kreise berechnet. Für die effiziente Bestimmung der Kreislinie verwendet das Exhaustive Pixel Placement einen leicht angepassten Algorithmus von Bresenham [5]. Der Vorteil von Bresenhams Verfahren ist, dass Ganzzahloperationen für die Berechnung der Kreislinie vollständig ausreichen. Der Pseudocode dieses Ansatzes ist in Algorithmus 3 aufgeführt.

Es reicht zur Bestimmung der Kreislinie völlig aus, nur einen Oktanten des Kreises zu berechnen, und die restlichen mittels Spiegelung zu erzeugen. Zusätzlich wird mit der Zählschleife in den Zeilen 5 bis 9 für eine Linienbreite von zwei Pixeln gesorgt. Diese Erweiterung des Bresenham - Algorithmus ist nötig, da der Algorithmus nicht garantiert, dass man einen Kreis nur mit inkrementeller Erhöhung des Radius vollständig füllen kann. In Abbildung 3.5 wird gezeigt, wie sich eine unterschiedliche Linienbreite auf das Pixel Placement auswirkt. In der linken Abbildung 3.5(a) wurde das Pixel Placement mit dem Standard - Bresenham - Algorithmus durchgeführt. Hierbei wurden einige Pixel nicht als mögliche neue Positionen verwendet, was sich in weißen Flecken bemerkbar macht. Gelöst wurde dieses Problem in Abbildung 3.5(b) mit einer Liniendicke von zwei Pixeln. Dadurch werden alle Pixel als mögliche Versetzungspositionen gefunden und in das Exhaustive Pixel Placement miteinbezogen.

Die besondere Eigenschaft des Exhaustive Pixel Placement ist, dass der Benutzer steuern

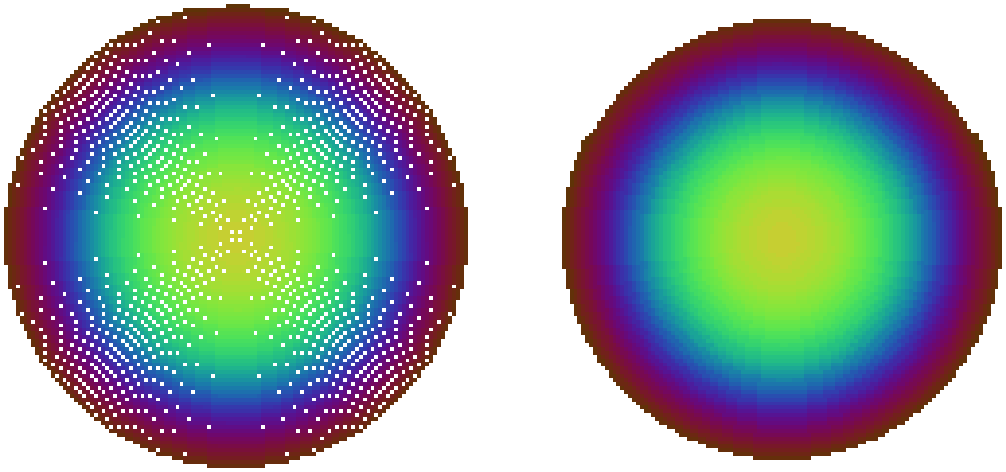
**Algorithmus 3:** Methode *calcNewPositions*
Berechnet nach Bresenham alle Punkte auf einer 2 Pixel dicken Kreislinie mit gegebenem Radius und Mittelpunkt

**Eingabe**: Mittelpunkt des Kreises *center*, Radius des zu zeichnenden Kreises
**Ausgabe**: Liste von Punkten, die auf einer zwei Pixel dicken Kreislinie liegen

```
1  int xEnd := (int) Math.round(Math.cos(0.25 * Math.PI) * radius);
2  int y := radius;
3  int e := 5 - 4 * radius;
4  for x := 0 to xEnd do
5      for i := 0 to 1 do
6          int xArray[ ] := x + center.x, x + center.x, −x + center.x, −x + center.x, y +
           center.x, y + center.x, −y + center.x, −y + center.x;
7          int yArray[ ] := y + center.y + i, −y + center.y + i, y + center.y + i, −y + center.y
           + i, x + center.y + i, −x + center.y + i, x + center.y + i, −x + center.y + i;
8          for counter := 0 to xArray.length do
9              result.add(new Point(xArray[counter], yArray[counter]));
10     if e <= 0 then
11         e += 8 * x + 12;
12     else
13         e += 8 * x - 8 * y + 20;
14         y - -;
15 return result;
```

kann, wie viele der überdeckenden Punkte verschoben werden. Der vom Benutzer angegebene Prozentwert, der aussagt, wie viel der ursprünglichen Überdeckung erlaubt ist, hat direkten Einfluss auf das Verfahren. Dieser steuert schließlich das Ausmaß des Pixel Placement, also wie viele überdeckende Punkte von ihrer Ausgangsposition verschoben werden. Der vom Benutzer gesetzte Prozentwert muss aber auf einen für den Algorithmus sinnvollen Wert umgerechnet werden. Zumal in den oben beschriebenen Methoden vorausgesetzt wurde, dass der Wert *maxAllowedOverlap* in irgendeiner Form schon bekannt ist. Dieser Wert muss pro Pixelposition angeben, wie viele Punkte sich an dieser Stelle überdecken dürfen.

Ferner darf die Umrechnung des Prozentwerts in den absoluten Wert *maxAllowedOverlap* nicht global geschehen, da eine Ungleichverteilung der Daten zu Problemen führen würde. Hierbei würde nämlich ein einziges Pixel mit sehr hoher Überdeckung dazu führen, dass bei allen anderen Koordinaten mit weniger Überdeckung, relativ gesehen, zu viele Punktverschiebungen durchgeführt werden. Der einzig richtige Ansatz ist also, den Wert *maxAllowedOverlap* nur lokal zu bestimmen. Es wäre nun möglich, ihn pro Pixel zu berechnen, jedoch wird aus Effizienzgründen der Bildraum in Gitterzellen unterteilt. In der Referenzimplementierung wurde eine experimentell gesetzte Anzahl von 200 Zellen verwendet. Anschließend wird pro Gitterzelle die Umrechnung des relativen Prozentwerts vorgenommen. Hierbei wird zunächst pro Zelle festgestellt, wie viele Punkte in ihr enthalten sind. Nun wird vereinfachend



(a) Berechnung der Kreispunkte bei einer Linienstärke von einem Pixel.

(b) Bei einer Lininendicke von zwei Pixeln, werden alle Pixel mindestens einmal von Bresenhams Algorithmus zurückgegeben.

Abbildung 3.5: Auswirkungen der Liniendicke beim Berechnen der Kreislinie auf das Pixel Placement. Im linken Bild sind die nicht verwendeten weißen Pixel deutlich zu erkennen. Mit einer Linienbreite von zwei Pixeln kann dieses Problem jedoch umgangen werden (rechte Grafik).

angenommen, dass sich alle Punkte in der Gitterzelle überdecken. Im Anschluss daran wird der Radius des Kreises berechnet, welcher alle sich überdeckenden Punkte beinhalten könnte. Dieser so berechnete Radius wird nun in Gleichung 3.4 mittels des vom Benutzer gesetzten Prozentwerts verringert.

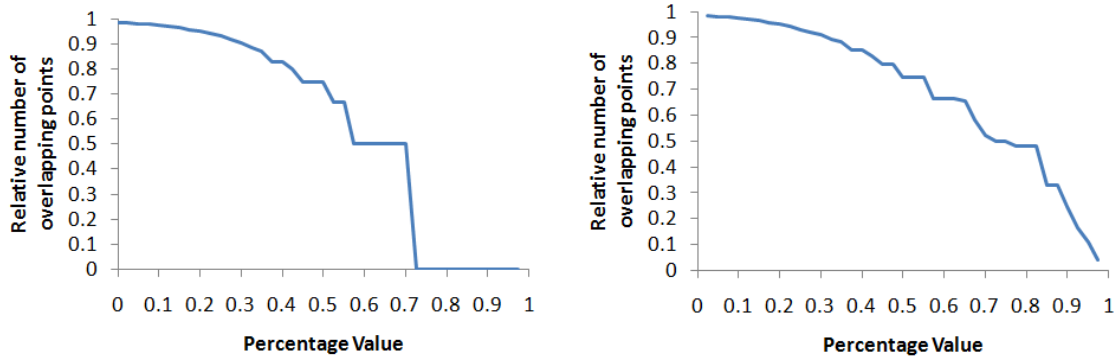
$$radius_{xPos,yPos} = percentageOverlapAllowed \cdot \sqrt{\frac{pointsPerCell_{xPos,yPos}}{\pi}} \tag{3.4}$$

Dabei ist in $pointsPerCell_{xPos,yPos}$ abgelegt, wie viele Datenpunkte in der entsprechenden Zelle enthalten sind. Daraufhin kann die maximal erlaubte Überdeckung in absoluten Zahlen berechnet werden. Hierzu wird in Gleichung 3.5 im Zähler bestimmt, wie viele Punkte in der Zelle enthalten sind. Im Nenner wird berechnet, wie viele Pixelkoordinaten der Kreis mit dem reduzierten Radius bereitstellt. Durch die Division wird nun die maximale Anzahl an Punkten pro Pixel ermittelt.

$$maxAllowedOverlap_{xPos,yPos} \approx \frac{pointsPerCell_{xPos,yPos}}{\pi \cdot radius^2_{xPos,yPos}} \tag{3.5}$$

Besonders entscheidend für die interaktive Steuerung durch den Benutzer ist, dass sich das Verfahren halbwegs linear bezüglich des gesetzten Prozentwertes verhält. Verwendet man in Gleichung 3.5 einfaches Runden, so addieren sich die Rundungsfehler auf und das Ergebnis ist eine eher treppenförmige Relation zwischen Prozentwert und Auswirkungen des Pixel Placements. In Abbildung 3.6(a) wird diese schlechte Relation gezeigt. Das Problem hierbei ist,

dass für viele große Prozentwerte immer das gleiche Ergebnis produziert wird. Das liegt daran, dass große Prozentwerte für $maxAllowedOverlap_{xPos,yPos}$ hauptsächlich einen Wert zwischen 1 und 2 annehmen.




(a) Beim einfachen Runden wird trotz verschiedener Prozentwerte dasselbe Ergebnis produziert

(b) Verbessertes Runden führt zu deutlich glatterer Kurve

Abbildung 3.6: Diese Grafiken zeigen den Einfluss der verschiedenen Rundungsarten des Wertes $maxAllowedOverlap_{xPos,yPos}$ auf das Pixel Placement - Verfahren. Auf der x-Achse wird jeweils der vom Benutzer gesetzte Prozentwert und auf der y-Achse die relative Anzahl überdeckender Punkte abgetragen.

Eine einfache Lösung dieses Problems war es, eine Art von intelligentem Runden einzuführen, welches beispielsweise den Wert 1.1 jedes zehnte Mal auf 2 rundet und alle anderen Male immer auf 1. Hierbei muss also pro Position abgespeichert werden, wie oft die Rundungsfunktion schon aufgerufen wurde. Die so geschaffene Rundungsfunktion löst das Problem zwar noch nicht vollständig (siehe Abbildung 3.6(b)), ist aber hinreichend gut, um sie bestehen zu lassen.

Die Laufzeit des hier vorgestellten Verfahrens hängt zum einen sehr stark vom Grad der Überdeckung und zum anderen vom zur Verfügung stehenden freien Platz ab. Gerade bei großen Datenmengen, die auf einem kleinen Display angezeigt werden sollen, kann das Verfahren von mehreren Sekunden bis zu wenigen Minuten in Anspruch nehmen. Beispielsweise benötigt das Verfahren ein bis zwei Sekunden, um 10 000 aufeinander liegende Punkte zu verschieben. Jedoch ist der unschlagbare Vorteil dieses Verfahrens, dass nach Anwendung des Algorithmus keine Punkte mehr aufeinander liegen. Trotzdem kann eine solche lange Berechnungsdauer der Interaktivität des Programms schaden, und daher werden im nächsten Abschnitt heuristische Verfahren mit linearer Laufzeit vorgestellt.

### 3.2.2 Heuristisches Pixel Placement

Das Hauptaugenmerk beim heuristischen Pixel Placement lag auf einer Laufzeitverbesserung zur besseren Unterstützung der interaktiven Programmnutzung. Jedoch darf ein solches heuristisches Verfahren nicht so viele visuelle Artefakte erzeugen, dass der Einsatz der Pixel Placement - Technik sinnlos wird.

Der Flaschenhals bei der Ausführung des Exhaustive Pixel Placement - Verfahrens liegt in der häufigen Berechnung von Kreislinien. Deswegen werden in den folgenden Techniken die zu verschiebenden Datenpunkte auf zufällige, in der Nähe liegende Positionen versetzt. Hierbei kann aber nicht garantiert werden, dass sich hinterher keine oder nur eine bestimmte Anzahl von Datenpunkte mehr überlappen. Ferner können Bereiche, die vom Exhaustive Pixel Placement - Algorithmus verwendet würden, bei den heuristischen Verfahren zufällig nicht besetzt werden, was sich besonders in Bereichen mit hoher Überdeckung bemerkbar macht.

Zum besseren Verständnis werden in den nachfolgenden Abschnitten die vorgestellten Techniken auf einen Testdatensatz angewendet und das jeweilige Resultat als Bild angeführt. Der hierbei eingesetzte Datensatz entspricht demjenigen, welcher auch schon in der Abbildung 3.4 verwendet wurde. Er besteht aus 155 Punkten mit denselben Koordinaten, wobei der Farbwert zufällig aus einer Gleichverteilung gezogen wurde und zwischen 0.0 und 1.0 liegt. Die Heuristik ist dabei umso besser, je mehr Ähnlichkeit des Ergebnisses zur Abbildung 3.4(c) besteht.

#### Verwendung einer Normalverteilung

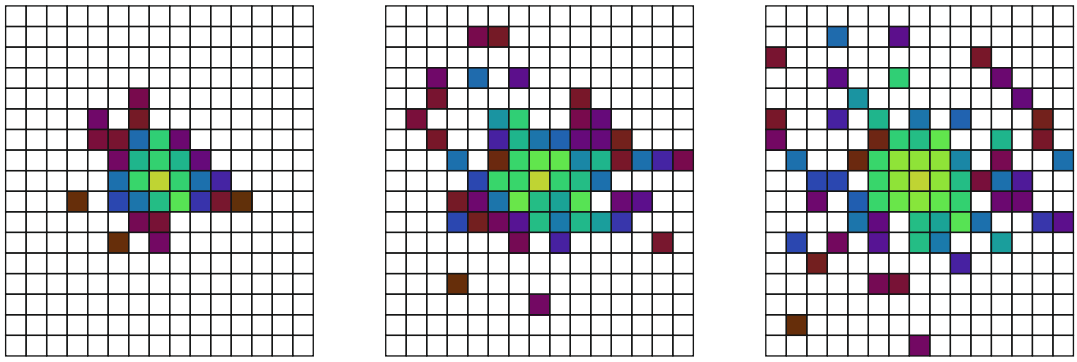
Statistikprogramme (wie beispielsweise R) bieten häufig die Möglichkeit, Daten künstlich zu verrauschen, um Regionen mit hoher Dichte besser erkennbar zu machen. Meistens werden die Daten dabei gemäß einer Normalverteilung um den Ursprungsort herum zufällig verteilt. Die erste Idee war nun, diese Technik leicht abgewandelt zu verwenden, um sich überdeckende Datenpunkte zu versetzen. Bereiche mit hoher Überdeckung sollten behoben werden, in dem Punkte gemäß einer Normalverteilung versetzt werden. Im Gegensatz zum einfachen Verrauschen der Daten wird hierbei jedoch auch noch Rücksicht auf die jeweilige Punktedichte und auf die Farbwerte genommen. In Abbildung 3.7 ist ein beispielhaftes Ergebnis dieser Technik zu sehen.

Die nun folgende Formel 3.6 zeigt die Berechnung einer neuen Koordinate (hier: die neue Position in der x-Achse). Die Berechnung der zweiten Koordinate ist identisch zu der hier gezeigten Berechnungsvorschrift und wird daher weggelassen.

$$\begin{aligned} xPos_{new} \approx\ & xPos_{orig} + randomValue_x \cdot colorValue_{normalized} \\ & \cdot \sqrt{noOfPointsAtPos_{xPos_{orig},yPos_{orig}}} \cdot userSetFactor \end{aligned} \tag{3.6}$$

Zur Berechnung der neuen Koordinaten wird somit eine zufällig aus einer Normalverteilung gezogene Zahl verwendet. Damit das Ergebnis nicht bei jedem Ausführen des Algorithmus anders aussieht, werden beim Einlesen der Daten schon vorsorglich pro Datenpunkt zwei normalverteilte Zufallszahlen im Intervall von $[-1,1]$ generiert. Als Parameter für die Normalverteilung wurden aus theoretischen und praktischen Gründen als Mittelpunkt $\mu = 0$ und als Standardabweichung $\sigma = 0.5$ gewählt.

Ferner fließt noch der Farbwert des jeweiligen Punktes – also der Wert des Datenpunktes in der dritten Dimension – ein. Dieser steuert nämlich, wie stark die Auslenkung nach außen ist. Hierzu wird der auf das Intervall von $[0,1]$ normalisierte Farbwert als Faktor in die Berechnung miteinbezogen. Zusätzlich sollen die Punkte umso weiter verstreut werden, je höher die



(a) 33 Prozent Pixel Placement (b) 66 Prozent (c) Maximale Versetzung der Datenpunkte

Abbildung 3.7: Heuristisches Pixel Placement nach einer Normalverteilung angewendet auf einen Testdatensatz mit 155 Punkten, welche alle dieselben Koordinaten haben. Hier wird nur ein Ausschnitt des Ergebnisses gezeigt, welcher jedoch – zur besseren Vergleichbarkeit – genau dem Ausschnitt der Abbildung 3.4 entspricht. In den einzelnen Teilgrafiken werden die Ergebnisse für verschiedene Einstellungen des $userSetFactor$ präsentiert.

Dichte an der Originalposition ist. Hierzu wird die Quadratwurzel der Anzahl der an dieser Stelle liegenden Punkte als weiterer Faktor verwendet. Die Quadratwurzel soll hierbei einen ähnlichen Radius wie beim Exhaustive Pixel Placement erzeugen. Letztendlich muss auch dieses Verfahren einen Einfluss des Benutzers auf die Stärke der Versetzungen ermöglichen. Um dies zu gewährleisten, wird der Wert $userSetFactor$ im Intervall von $[0, 1]$ eingesetzt. Falls der Wert durch den Benutzer beispielsweise auf den Wert 0 gesetzt wird, so hat der rechte Summand in der Formel 3.6 keinerlei Einfluss auf die neue Position des Datenpunkts. Der Einfluss des Parameters $userSetFactor$ auf das Verfahren kann in Abbildung 3.7 beobachtet werden.

#### Versetzung von Punkten nach einer Gleichverteilung

Nachdem augenscheinlich das oben vorgestellte heuristische Verfahren dem Ergebnis des Exhaustive Pixel Placements nicht entspricht, wird in diesem Abschnitt ein besseres Verfahren beschrieben. Das Problem bei der obigen Berechnungsvorschrift war der Ansatz an und für sich. Die Normalverteilung sorgt schließlich gerade dafür, dass die Punktedichte in der Nähe der Originalposition am höchsten ist. Das Ziel des Pixel Placement - Verfahrens sollte jedoch sein, eine Gleichverteilung der Daten um den Ausgangspunkt herum zu schaffen. Ferner sollte wiederum die Punkteüberdeckung und der Farbwert Einfluss auf die Versetzung eines jeden einzelnen Punktes haben.

Die grundsätzliche Idee dieses Ansatzes ist, für jeden Punkt seine Kreisbahn um den ursprünglichen Punkt herum zu finden. Diese kann einfach durch die Anzahl der schon platzierten Punkte bestimmt werden. Dabei wird jedoch vorausgesetzt, dass die Punkte nach Farbwert sortiert vorliegen. Schließlich werden die Punkte in der gegebenen Reihenfolge sequenziell

abgearbeitet. Liegt nun keine Sortierung nach Farbwerten vor, so können visuelle Artefakte entstehen. Anschließend wird auf dieser Kreisbahn der Punkt gemäß einer Gleichverteilung verteilt.

Die Berechnung der Kreisbahn beziehungsweise des Radius des Kreises, auf dem der Punkt liegen soll, wird in der folgenden Formel 3.7 durchgeführt. Hierbei wurde im ersten Schritt bewusst auf das Kürzen verzichtet, um ein besseres Verständnis zu fördern.

$$radius = \sqrt{\frac{noOfElemAtPos}{\pi}} \cdot \sqrt{\frac{currentNumberOfElemAtPos}{noOfElemAtPos}} \cdot userSetFactor \tag{3.7}$$

$$= \sqrt{\frac{currentNumberOfElemAtPos}{\pi}} \cdot userSetFactor \tag{3.8}$$

Der erste Faktor in Formel 3.7 ist aus der Umformung der Gleichung $A = \pi r^2$ entstanden und beschreibt, wie groß der Kreisradius sein müsste, um alle überdeckenden Punkte der Ursprungsposition nicht überdeckend darzustellen. Der zweite Faktor liegt im Intervall von $[0,1]$. Er gibt an, wie groß der Radius relativ sein muss, um allen bisher verarbeiteten Datenpunkten eine Position ohne Überdeckung zuzuweisen. Hierbei steht der Wert *currentNumberOfElemAtPos* für die Anzahl an Punkten mit denselben Ursprungskoordinaten, die schon abgearbeitet wurden. Die Quadratwurzel beim zweiten Faktor ist notwendig, da der Flächeninhalt von Kreisen bei zunehmendem Radius quadratisch ansteigt und dies wieder ausgeglichen werden muss. Schlussendlich muss der Benutzer auch bei diesem Verfahren die Möglichkeit haben, Einfluss auf die Stärke des Pixel Placements zu nehmen. Folglich findet sich in der Berechnungsvorschrift der Wert *userSetFactor*, welcher im Intervall von $[0,1]$ liegt. Weist der Benutzer beispielsweise dem Parameter den Wert 0 zu, so wird der Radius auch 0 sein. Die Auswirkungen verschiedener Werte für die Variable *userSetFactor* kann in den Abbildungen 3.8(a), 3.8(b) und 3.8(c) betrachtet werden.

Die Formel in 3.8 beschreibt genau denselben, oben beschriebenen Sachverhalt. In ihr wurde nur von der Möglichkeit Gebrauch gemacht, den Wert *noOfElemAtPos* aus der Formel zu kürzen, um eine einfachere und effizientere Berechnung zu erreichen.

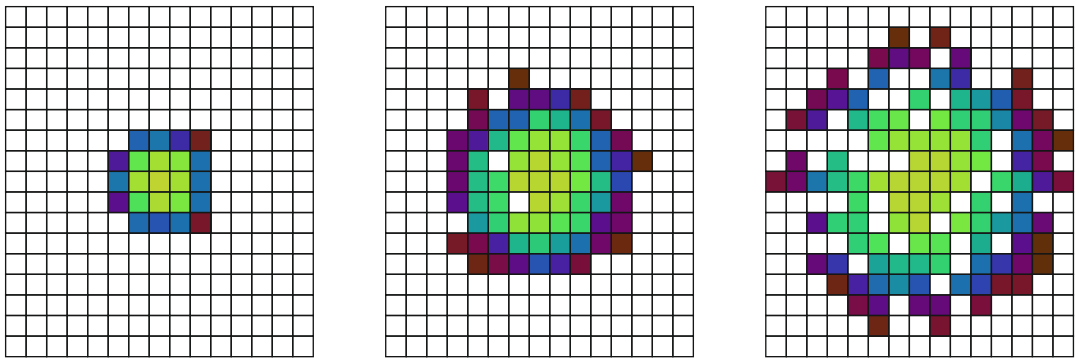
Die zweite Stufe des Verfahrens ist nun, die Richtung der Versetzung zu bestimmen; also die Position auf der Kreisbahn. Hierzu könnte beispielsweise eine Zufallszahl aus einer Gleichverteilung im Intervall von $[-\pi,\pi)$ gezogen werden. Da aber schon für das obige heuristische Verfahren pro Datenpunkt zwei normalverteilte Zufallszahlen zur Verfügung stehen, wird in der Referenzimplementierung der Winkel aus diesen beiden mittels des Arkustangens bestimmt. Dieser Schritt wird in Formel 3.9 gezeigt. Der daraus resultierende Winkel folgt einer Gleichverteilung im Intervall von $[-\pi,\pi]$.

$$angle = 2 \cdot \arctan\left(\frac{randYNormDist}{randXNormDist}\right) \tag{3.9}$$

Abschließend muss die neue Position des Punktes aus dem berechneten Radius und der Richtung bzw. Position auf der Kreisbahn bestimmt werden. Dies geschieht in den Formeln 3.10 und 3.11. Dabei wird ein Produkt aus dem Radius mit dem Kosinus beziehungsweise Sinus des zufälligen Winkels gebildet und zur Originalkoordinate addiert.

$$xPos_{new} = xPos_{orig} + radius \cdot \cos(angle) \tag{3.10}$$
$$yPos_{new} = yPos_{orig} + radius \cdot \sin(angle) \tag{3.11}$$

In Abbildung 3.8 wird das Ergebnis des hier vorgestellten heuristischen Verfahrens gezeigt. Hierfür wurde der oben beschriebene Testdatensatz bestehend aus 155 Punkten verwendet. Deutlich sichtbar sind die vielen weißen Lücken innerhalb der Kreisfläche, welche aus der zufälligen Verteilung der Punkte resultieren.



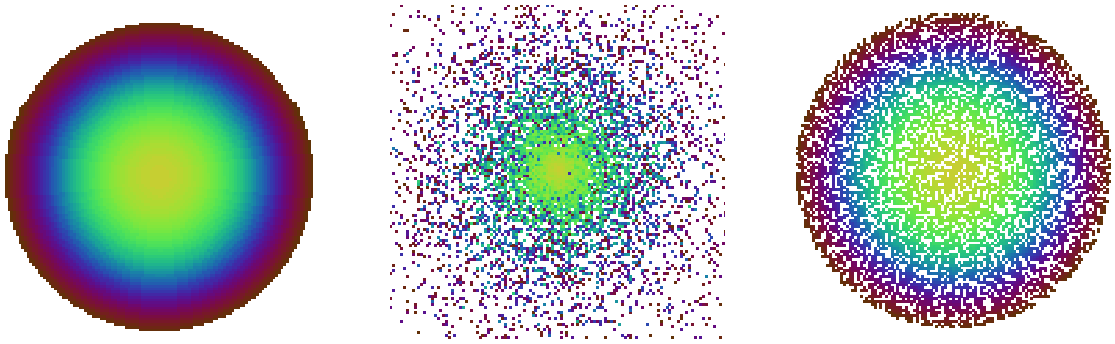
(a) 33 Prozent Pixel Placement (b) 66 Prozent (c) Maximale Versetzung der Datenpunkte

Abbildung 3.8: Ergebnis des heuristischen Pixel Placements, das die Datenpunkte mittels einer Gleichverteilung versetzt. Angewendet wurde das Verfahren auf einen Datensatz mit 155 Punkten, welche alle dieselben Koordinaten haben. In den einzelnen Grafiken werden die Ergebnisse für verschiedene Werte des Parameters *userSetFactor* gezeigt.

### 3.2.3 Vergleich der vorgestellten Techniken

In diesem Abschnitt werden die drei oben beschriebenen und vorgestellten Ansätze miteinander verglichen. Dabei werden sowohl die Laufzeit der Algorithmen als auch die Güte der heuristischen Verfahren untersucht. Zum einfachen Vergleich des Auftretens von visuellen Artefakten werden in Abbildung 3.9 die verschiedenen Techniken auf denselben Datensatz angewendet. Der Datensatz besteht aus 10 000 Punkten mit denselben Koordinaten und zufällig erzeugten, gleichverteilten Farbwerten im Intervall von $[0,1]$.

Wie in Abbildung 3.9(b) deutlich ersichtlich ist, erzeugt das normalverteilte, heuristische Pixel Placement die stärksten visuellen Artefakte. Ferner hat dieses Verfahren den Nachteil, dass die Streuung der Punkte zu groß ist. Anders als bei den beiden anderen Verfahren, reicht der dargestellte Bereich nicht aus, um alle Datenpunkte unterzubringen. Zusätzlich zu der schlechten Platzausnutzung ist die Punktedichte in der Nähe der Originalposition viel höher als weiter außen. Allein auf Grund der starken visuellen Unterschiede zum nicht heuristischen



(a) Exhaustive Pixel Placement

(b) Heuristisches Pixel Plaxement — Normalverteilung

(c) Heuristisches Pixel Plaxement — Gleichverteilung

Abbildung 3.9: In dieser Grafik werden die verschiedenen, vorgestellten Pixel Placement - Techniken zur besseren Vergleichbarkeit auf ein und denselben Datensatz angewendet. Der Datensatz besteht aus 10 000 Punkten, die alle dieselben Koordinaten haben. In allen drei Bildern wird derselbe Ausschnitt gezeigt, was im mittleren Bild dazu führt, dass nicht alle Datenpunkte im sichtbaren Bereich liegen.

Verfahren kann diese Technik als Ersatz für das Exhaustive Pixel Placement ausgeschlossen werden.

Dahingegen kommt das zweite heuristische Verfahren, welches die Punkte um die Ursprungskoordinaten herum gleichverteilt anordnet, dem Exhaustive Pixel Placement schon sehr nahe. Die Farbverteilung der Punkte ähnelt sich sehr stark und auch die Größen der Kreise sind nahezu identisch. Einzig die weißen Stellen, welche durch die zufällige Platzierung der Datenpunkte entstehen, sind als visuelle Artefakte deutlich erkennbar. Jedoch zerstören die Artefakte nicht den Gesamteindruck; allein anhand der Grafik 3.9(c) kann man sich schon vorstellen, wie das Ergebnis des Exhaustive Pixel Placements aussehen würde.

Die Existenzgrundlage der heuristischen Verfahren beruht auf der langsamen Laufzeit des Exhaustive Pixel Placement - Algorithmus. Zur einfacheren Vergleichbarkeit der Laufzeiten wurden in der folgenden Tabelle die vorgestellten Techniken sowohl auf unterschiedlich große Testdatensätze als auch auf einen realen Datensatz angewendet. Die absoluten Zahlen, die auf einem 1,2 GHz Dual-Core Computer mit 2 Gigabyte Arbeitsspeicher erfasst wurden, sind dabei nicht so interessant wie die relativen Verhältnisse der Messwerte zueinander.

| | Exhaustive | Normalverteilung | Gleichverteilung |
|---|---|---|---|
| Testdaten (155 Punkte) | 0.0344 | 0.0069 | 0.0108 |
| Testdaten (10 000 Punkte) | 1.3300 | 0.0732 | 0.0755 |
| Testdaten (100 000 Punkte) | 40.4607 | 0.6748 | 0.7982 |
| Realdaten (37 788 Punkte) | 3.2544 | 0.2799 | 0.3097 |

Tabelle 3.1: Laufzeiten der verschiedenen Algorithmen gemessen in Sekunden

Wie schon zu erwarten war, ist das Exhaustive Pixel Placement mit Abstand das langsamste

Verfahren. Der Zeitunterschied liegt dabei etwa im Bereich von ein und zwei Größenordnungen. Der Unterschied wird dabei umso stärke, je mehr Punkte sich gegenseitig überdecken: Im dritten Testdatensatz, bei dem 100 000 Punkte dieselben Koordinaten haben, benötigt das Exhaustive Pixel Placement - Verfahren circa 40 Sekunden, während die beiden heuristischen Techniken noch unter einer Sekunde bleiben. Aber auch beim Datensatz aus der realen Welt ist der Unterschied um den Faktor 10 noch deutlich erkennbar. Schließlich ist der Grad an Überdeckung auch bei diesem Datensatz stark ausgeprägt und für die Zeitmessung wurden die Datenpunkte natürlich nicht verzerrt. Der Datensatz wird noch im Kapitel 4 eingehender beschrieben und vorgestellt.

Die verhältnismäßig langsame Ausführungsgeschwindigkeit der Exhaustive Pixel Placement - Technik hat mehrere Gründe, auf die in den folgenden Absätzen näher eingegangen wird:

Als erstes wird allein schon die Kreisberechnung bei hohem Überdeckungsgrad immer aufwendiger. Dies resultiert aus der linearen Abhängigkeit des Kreisumfangs vom Radius. Je größer die Überdeckung, desto größer ist der maximal zu verwendende Radius. Ein großer Radius führt wiederum zu einem hohen Berechnungsaufwand, da viele Punkte Teil der Kreislinie sind.

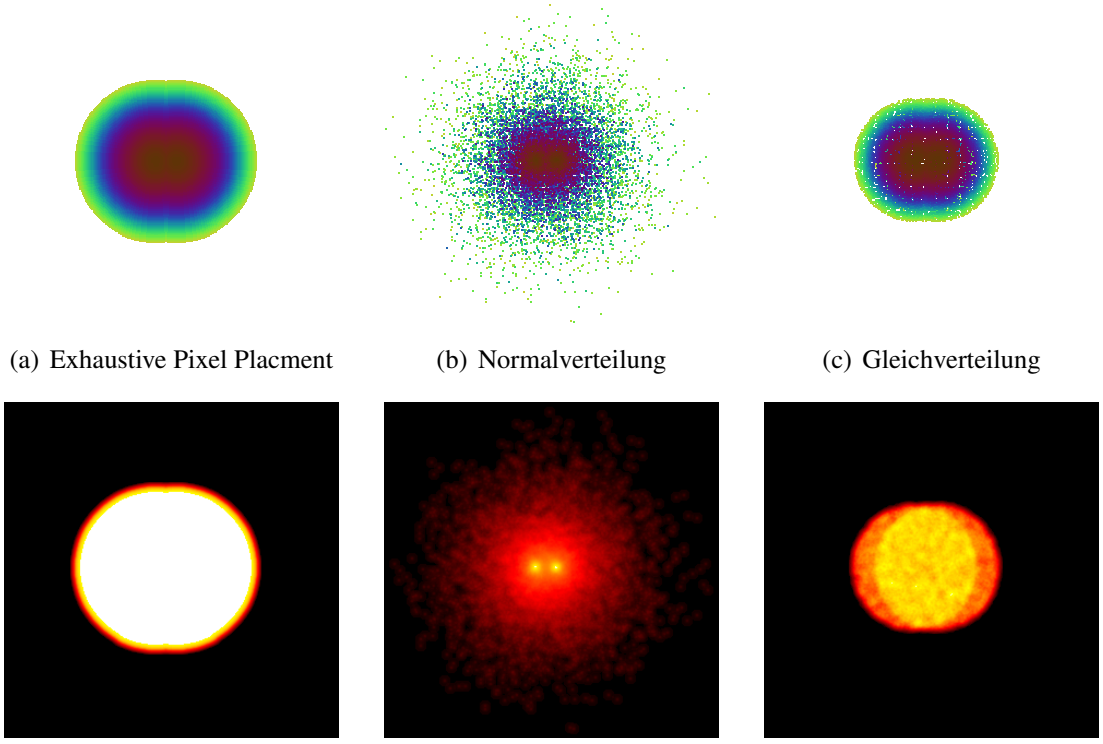
Zusätzlich wird die Kreisberechnung bei einem hohen Grad an Überdeckung auch sehr häufig aufgerufen. Nach dem oben beschriebenen Verfahren wird Bresenhams Algorithmus bei hundertprozentigem Pixel Placement für jeden Datenpunkt mindestens einmal ausgeführt, auch wenn dies eigentlich nicht notwendig ist. Eine Verbesserungsmöglichkeit wäre hierbei die vorherige Berechnung aller Kreise mit allen für den entsprechenden Datensatz denkbaren Radien mit dem Ursprung als Mittelpunkt. Die Kreise könnten zur Laufzeit einfach an die jeweilige Originalposition des Punktes verschoben werden und so die möglichen Versetzungspositionen liefern. Dies würde zwar den Speicherplatzbedarf des Algorithmus erheblich erhöhen, aber die Laufzeit des Exhaustive Pixel Placements sehr verkürzen.

Der letzte große Einflussfaktor auf das Pixel Placement - Verfahren ist der Überdeckungsgrad des jeweiligen Datensatzes. Hierbei spielt die relative Anzahl an überdeckenden Punkten eher keine Rolle. Viel entscheidender ist, ob sich die Überdeckung vor allem auf einzelne Regionen – oder noch schlimmer: einzelne Pixelpositionen – konzentriert. Gerade bei einzelnen sehr stark überdeckenden Positionen, oder kleinen Bereichen mit hoher Punktedichte müssen sehr viele Kreise auf freie Stellen hin überprüft werden.

Beim Vergleich der jeweiligen Laufzeiten fällt auf, dass die Normalverteilungsheuristik ein wenig schneller zu sein scheint als das andere heuristische Verfahren. Dies liegt wahrscheinlich daran, dass bei dem Gleichverteilungsverfahren mehr Berechnungen durchgeführt werden müssen. Schließlich muss hier nicht nur der jeweilige Radius ausgerechnet werden, sondern auch noch die zufällige Position auf der Kreisbahn.

Der große und wirklich schmerzhafte Nachteil der heuristischen Verfahren ist die nicht vorhandene Überprüfung auf Überdeckung. Sollten im schlimmsten Fall beispielsweise zwei nah benachbarte Pixelpositionen eine hohe Überdeckung aufweisen, so wird dies von den Heuristiken komplett ignoriert. In Abbildung 3.10 wurde ein speziell generierter Testdatensatz verwendet. Bei diesem überdecken sich an zwei nah benachbarten Positionen jeweils 10 000 Datenpunkte. Um einen tieferen Einblick in die Algorithmen zu erlangen, wurden nicht nur die Streudiagramme gezeigt, sondern auch die Dichteverteilungen mittels Farben visualisiert.

Der hierbei verwendete Colormap reicht von Schwarz und Rot für niedrige Dichte bis hin zu Gelb und Weiß, welche für Regionen mit hoher Dichte stehen.



(a) Exhaustive Pixel Placment (b) Normalverteilung (c) Gleichverteilung

(d) Dichteverteilung des Exhaustive Pixel Placment (e) Dichteverteilung der Normalverteilung (f) Dichteverteilung der Gleichverteilung

Abbildung 3.10: Diese Abbildung zeigt, dass die vorgestellten heuristischen Verfahren keine Überprüfung auf Überdeckung beim Versetzen durchführen. Für die vorliegende Grafik wurde ein Testdatensatz generiert, bei dem sich an zwei nah benachbarten Positionen jeweils 10 000 Datenpunkte überdecken. Zur besseren Darstellung wurde in der zweiten Zeile die jeweilige Dichtevisualisierung abgebildet. Der verwendete Colormap reicht von Schwarz / Rot (niedrige Dichte) bis hin zu Gelb / Weiß (hohe Dichte).

Die Heuristiken agieren nur lokal und überprüfen beim Versetzen nicht, wie viele andere Datenpunkte schon auf dem Pixel positioniert sind. Auch in Abbildung 3.10 wird die ähnliche Form von Exhaustive Pixel Placement und Gleichverteilungsheuristik sichtbar. Jedoch stimmen diesmal die Radien nicht überein, was wie schon erwähnt an dem heuristischen Vorgehen liegt. Deutlich sichtbar wird dieses Vorgehen anhand der Dichtevisualisierungen in den Abbildungen 3.10(d) und 3.10(f). Während das Exhaustive Pixel Placement in Abbildung 3.10(d) eine gleichmäßige Dichteverteilung auszeichnet, ist bei der Gleichverteilungsheuristik in Abbildung 3.10(f) eine Überlagerung zweier Kreise sichtbar. Diese Überlagerung, erkennbar an der gelben Doppelsichel, zeigt, dass Regionen mit Überdeckung voneinander völlig unabhängig abgearbeitet werden.

Trotz der Nachteile der Heuristik, welche die Punkte gleichverteilt versetzt, sollte man eine Kombination der Techniken anstreben. Um einen hohen Grad an Interaktion zu ermöglichen, sollte die Heuristik eine Art Vorschaubild erzeugen. Hierdurch kann der Benutzer schnell erkennen, wie stark er Verzerrung und eben auch Pixel Placement auf den Datensatz anwenden sollte, um die bestmögliche Visualisierung der Daten zu erreichen. Nachdem der Benutzer mit den Einstellungen zufrieden bzw. der Computer nicht mehr ausgelastet ist, kann das aufwendigere Exhaustive Pixel Placement durchgeführt werden. Somit wird ein hoher Grad an Interaktivität ermöglicht, und trotzdem ist das Zwischenergebnis dem Endresultat ähnlich genug, um den Benutzer bei der visuellen Datenanalyse zu unterstützen.

## 3.3 Optimierungsproblem

Typischerweise ist der Benutzer an einer Visualisierung interessiert, die sowohl noch die ursprüngliche Datenverteilung erkennen lässt als auch interessante Bereiche hervorhebt. Dies ist natürlich immer ein Balanceakt zwischen zu großer Verzerrung und zu schlechter Vergrößerung interessanter Bereiche. Zusätzlich darf bei Streudiagrammen nicht vergessen werden, dass die Überdeckung von Punkten zu Fehlinterpretationen der Daten führen kann. Aber zu starke Verzerrung kann genauso gut zu Fehlinterpretationen führen. Das Auffinden der optimalen Ansicht in Streudiagrammen gleicht also einem Optimierungsproblem, welches der Benutzer lösen will.

Falls man nun die Güte einer Ansicht auf die Daten – generiert durch die Generalized Scatter Plots (inklusive Verzerrung und Pixel Placement) – bewerten will, so müssen zwei Einflussfaktoren bzw. Optimierungsziele berücksichtigt werden.

Das erste Optimierungsziel ist eine möglichst originalgetreue Darstellung der Datenpunkte. Dies ist wichtig, um das erzeugte Streudiagramm zu verstehen und es analysieren zu können. Das hier aufgeführte Ziel kann auch mathematisch greifbar gemacht werden: Für eine gegebene Menge von $n$ Punkten $p_1,...p_n$ soll $O(p_i)$ die Originalposition und $N(p_i)$ die berechnete Position im erzeugten Streudiagramm beschreiben. Zusätzlich sei eine Distanzfunktion $d(O,N)$ definiert, welche im Streudiagramm die euklidische Distanz zwischen den Punkten $O$ und $N$ berechnet. Das Maß zur Bestimmung des Versetzungsfehlers (*displacement error*) ist in Gleichung 3.12 aufgeführt.

$$e_{disp} = \sum_{i=1}^{n} \frac{d\left(O(p_i), N(p_i)\right)}{n} \tag{3.12}$$

Der Versetzungsfehler misst die Stärke von Positionsänderungen aller Datenpunkte zwischen dem ursprünglichen Streudiagramm und dem erzeugten Generalized Scatter Plot. Das zweite Optimierungsziel ist eine möglichst geringe Anzahl an sich überdeckenden Punkten. Der Überdeckungsfehler kann auch in Zahlen erfasst werden, und ist in Gleichung 3.13 beschrieben.

$$e_{overlap} = \frac{\left|\{p_i | \exists j : N(p_i) = N(p_j) \wedge i \neq j\}\right|}{n} \tag{3.13}$$

Hierbei muss beachtet werden, dass zwischen den beiden Optimierungszielen ein Konfliktpotential besteht. Falls eine starke Verzerrung des Datenraums durchgeführt wird, erreicht man üblicherweise einen niedrigeren Überdeckungsfehler. Jedoch führt eine stärkere Verzerrung gleichzeitig zu einem höheren Versetzungsfehler. Um eine kombinierte Optimierungsfunktion zu erhalten, schlägt diese Arbeit eine gewichtete Summe der beiden einzelnen Fehlerfunktionen vor. Das Ziel der in Gleichung 3.14 vorgestellten Optimierungsfunktion muss sein, einen möglichst geringen Gesamtfehler zu produzieren.

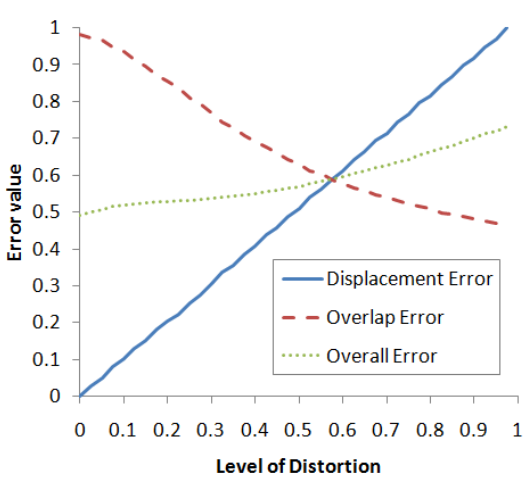
$$c \cdot e_{disp} + (1 - c) \cdot e_{overlap} \rightarrow \text{MIN} \tag{3.14}$$

In dieser Gleichung ist $c$ eine Proportionalitätskonstante im Intervall $[0, 1]$ zur Steuerung der Einflüsse der oben vorgestellten Optimierungsziele. Eine Erhöhung von $c$ führt zu einer stärkeren Bestrafung des Versetzungsfehlers und zu einer geringeren von Punktüberdeckungen. Bei einer Verringerung der Konstante würde das Umgekehrte passieren. Für ein ausbalanciertes Gewicht von Versetzungs- und Überdeckungsfehlern sollte c auf 0.5 gesetzt werden.

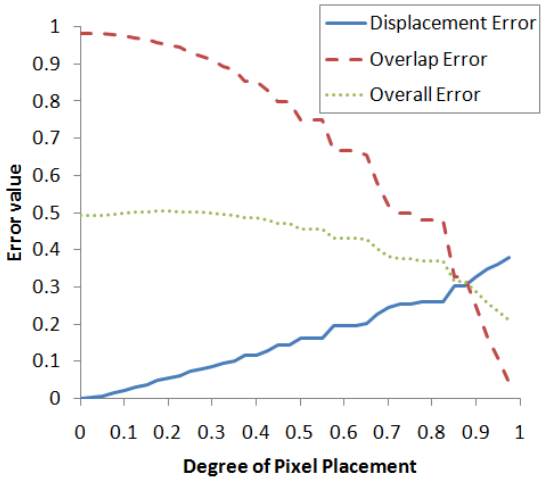
Die Ergebnisse beider Fehlerfunktionen wurden für verschiedene Einstellungen für Verzerrung und Pixel Placement in Abbildung 3.11 ausgewertet. Hierfür wurde der Telefondatensatz aus Kapitel 4.1 verwendet. In der ersten Analyse wurde nur der Verzerrungsgrad verändert, um festzustellen, wie sich die Verzerrung auf beide Optimierungsziele auswirkt. Die Resultate dieser Analyse sind in Abbildung 3.11(a) abgebildet. Während mit zunehmendem Verzerrungsgrad der Versetzungsfehler ansteigt, sinkt der Überdeckungsfehler gleichzeitig. Bis zu einer vierzig prozentigen Verzerrung bleibt der Gesamtfehler im Intervall $[0.5, 0.55]$.

Die Variation der zugelassen Überdeckung bzw. des durchgeführten Pixel Placements hat einen ähnlichen Einfluss auf die vorgestellten Fehlerfunktionen. Wie in Abbildung 3.11(b) erkennbar, hat das Pixel Placement verglichen mit dem Verzerrungsgrad einen geringeren Einfluss auf den Versetzungsfehler. Der optimale Grad an Pixel Placement liegt für den hier verwendeten Datensatz bei 100 Prozent. Alle hier abgebildeten Ergebnisse und Berechnungen basieren auf gleichen Gewichten für beide Fehlerfunktionen ($c = 0.5$). Für diesen Datensatz wurde die Proportionalitätskonstante $c$ auf 0.5 gesetzt, da sowohl der Versetzungsfehler als auch der Überdeckungsfehler in diesem Fall als gleich wichtig angesehen wurden.

Viel interessanter und relevanter für den realen Einsatz ist jedoch die gleichzeitige Variation von Verzerrung und Pixel Placement. In der Abbildung 3.12 wird der Gesamtfehler für jede Konstellation aus Verzerrung und Pixel Placement gezeigt. Der Verzerrungsgrad wurde auf die x-Achse und der Grad an Pixel Placement auf die y-Achse gelegt. Der Gesamtfehler wird mittels verschiedenen Farben einer Heatmap visualisiert, wobei gelb für niedrige Werte steht und rot für hohe Fehlerwerte. Die Abbildung zeigt eine komplexe Interaktion zwischen Verzerrung und Pixel Placement, wobei die optimale Kombination der Werte bei (Verzerrung = 0.26, Pixel Placement = 1.0) liegen. Um den optimalen Punkt zu finden, wurde diejenige Kombination gesucht, welche einen minimalen Gesamtfehler besitzt. Schließlich soll das Ergebnis allen Verzerrungen zum Trotz dem Originalbild möglichst ähnlich sehen. Diese so gefundene beste Kombination kann zur Unterstützung des Benutzers als initiale Einstiegsvisualisierung verwendet werden. Das graphische Ergebnis der optimalen Parametereinstellungen wird in Abbildung 4.1 (zweite Reihe, rechtes Bild) gezeigt.




(a) Optimierungsfunktionen in Abhängigkeit vom Verzerrungsgrad




(b) Optimierungsfunktionen in Abhängigkeit von der Stärke des Pixel Placements

Abbildung 3.11: Die definierten Fehlerfunktionen in Abhängigkeit von Verzerrung bzw. Pixel Placement.

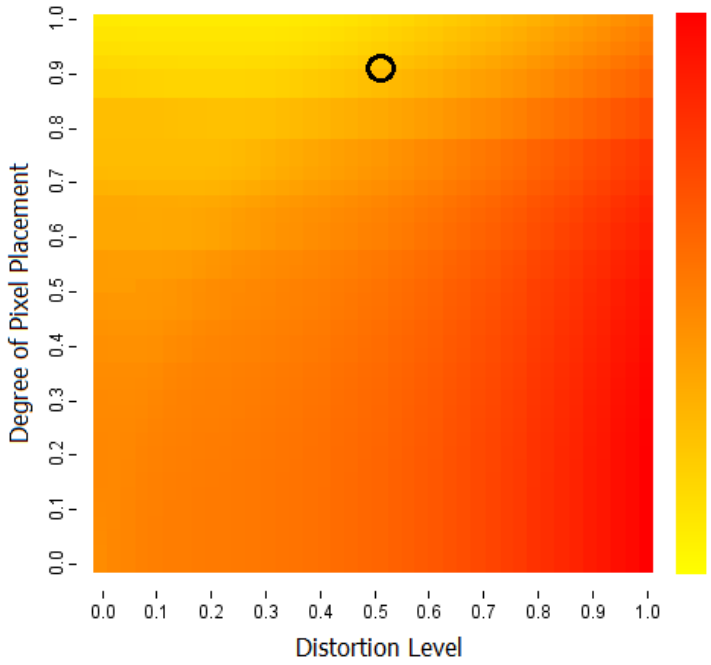



Abbildung 3.12: Der kombinierte Fehler in Abhängigkeit von Verzerrungsgrad und Stärke an Pixel Placement. Gelb repräsentiert niedrige und rot hohe Fehlerwerte. Die optimale Kombination der Parameter für gleich gewichtete Fehler ($c = 0.5$) wird durch einen Kreis gekennzeichnet. Der optimale Wert (niedrigster Wert in beiden Dimensionen mit kleinstem Fehler) kann sich in Abhängigkeit vom Gewicht $c$ verschieben.

## 3.4 Streudiagramm - Matrizen

In dem einführenden Kapitel wurde schon beschrieben, wie man mit Streudiagrammen mehr als nur zwei beziehungsweise drei Dimensionen gleichzeitig darstellen kann. Üblicherweise generiert man für jedes Paar von Dimensionen ein Streudiagramm und ordnet diese matrixartig an. Für die Generalized Scatter Plots bietet sich natürlich die gleiche Vorgehensweise an. Jedoch kann man hierbei eine Eigenheit der Streudiagramm - Matrizen ausnutzen. In diesen Matrizen herrscht eine Redundanz vor, da dasselbe Dimensionspaar zweimal verwendet wird, bloß jeweils an der Hauptdiagonalen gespiegelt. Daher zahlt es sich aus, in der oberen Diagonalhälfte die Generalized Scatter Plots wie gewohnt in der gewünschten Verzerrung und gewünschten Menge an Pixel Placement darzustellen. In der unteren Diagonalhälfte jedoch kann das traditionelle Streudiagramm eingetragen werden, was einen zusätzlichen Bezug zwischen verzerrter Ansicht und der ursprünglichen Darstellung herstellt.

Zur besseren Veranschaulichung dieses Vorschlags wurde er in Abbildung 3.13 auf einen Datensatz zur Überwachung der Serverperformanz angewendet. Sinn der Datenanalyse ist es hierbei herauszufinden, welche Systemressource mit einer hohen Systembelastung einhergeht. Die Streudiagramm - Matrix hilft dabei, den Überblick über die verschiedenen Dimensionen zu behalten und gleichzeitig die stark korrelierten Dimensionspaare zu finden. Bei der Abbildung 3.13 wurde ein mittlerer Verzerrungsgrad verwendet und keine Punktüberdeckung zugelassen.

Gerade durch die gleichzeitige Visualisierung der Ausgangsgraphen zusammen mit den verzerrten Varianten kann der Bezug auch bei statischen Bildern ohne Interaktionsmöglichkeit hergestellt werden. Der Vorteil hierbei ist, dass der Datenanalyst einen visuellen Vergleich der Korrelationen über mehrere Attribute hinweg in einer einzigen Ansicht vornehmen kann, ohne durch Überdeckungsprobleme – auch bei großen Datenmengen – daran gehindert zu werden. Beispielsweise könnte der Systembetreuer daher in diesem Fall den Flaschenhals im System aufspüren und präventive Maßnahmen ergreifen.

## 3.5 Referenzimplementierung

Ziel der Implementierung war es, ein umfassendes Rahmenwerk für verschiedene Verzerrungen und Pixel Placement - Techniken zu schaffen. Es sollten nicht nur einzelne Verzerrungstechniken auf die Daten angewendet werden, sondern der Benutzer sollte aus den Synergieeffekten mehrerer Verzerrungen profitieren. Aus diesem Grund wurde eine wohldefinierte Schnittstelle für das abstrakte Grundgerüst einer Verzerrung erstellt, welche eine einfache Erweiterbarkeit des Prototypen ermöglicht. Hierbei wurden drei verschiedene Verzerrungstechniken integriert: das HistoScale-, MultiRadial- und MultiAngular- Verfahren. Zusätzlich wurden verschiedene Sichten auf die Daten implementiert, welche die Datenanalyse effektiver gestalten sollen. Hierzu gehören verschiedene dichtebasierte Ansichten, beispielsweise Dichtegradienten oder Isolinien. Selbstverständlich wurde auch die Möglichkeit gegeben, die Farbskalierung zu ändern beziehungsweise die Datenpunkte zur Dichtevisualisierung mit halb transparenten gefüllten Kreisen zu umgeben. Außerdem kann das Programm Polygone einlesen und auch verzerrt darstellen, was gerade für geographische Anwendungen interessant

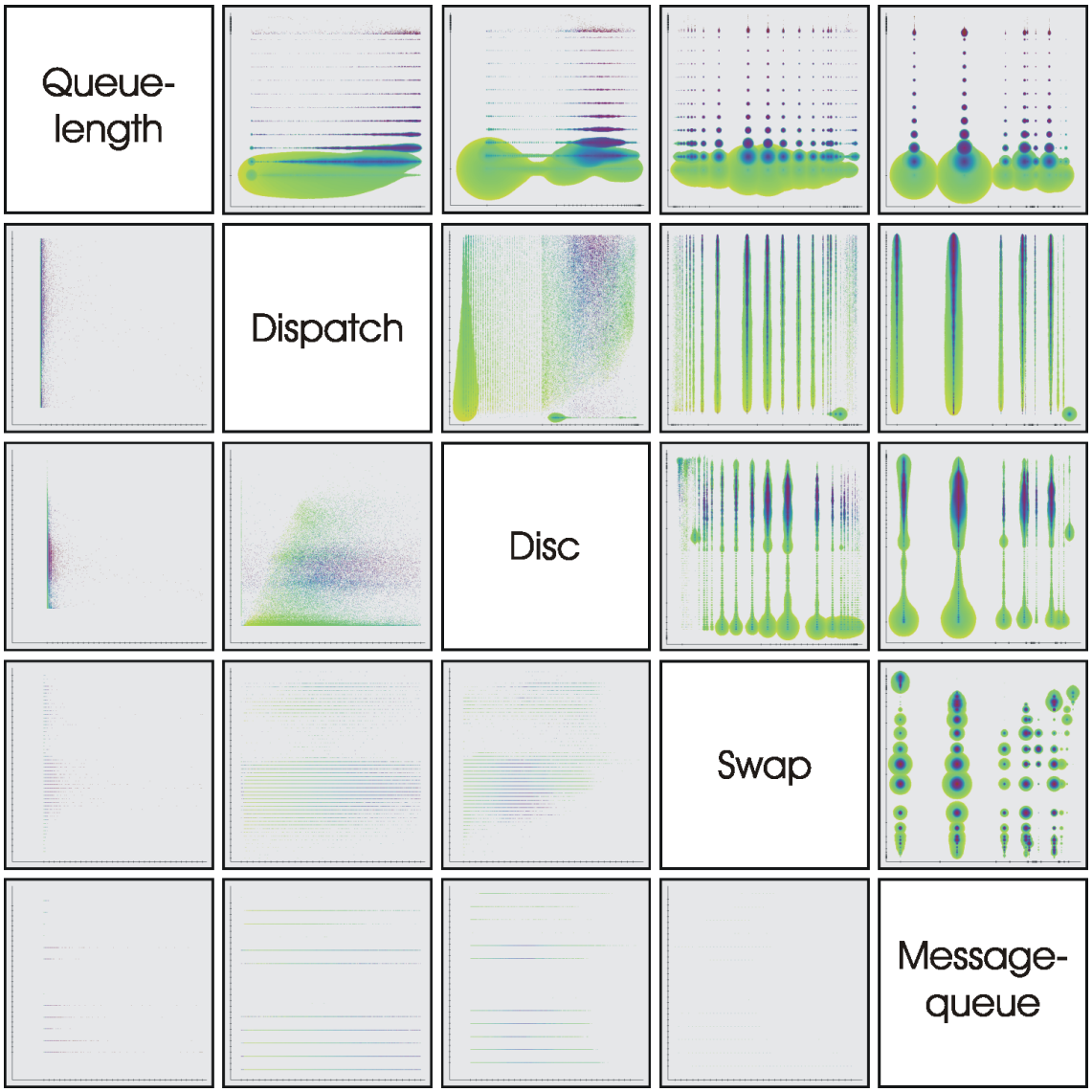



Abbildung 3.13: Streudiagramm - Matrix zur gleichzeitigen Visualisierung mehrerer Dimensionen. Die Farbe repräsentiert die Auslastung des Servers in Prozent. Als Ergänzung zur normalen Streudiagramm - Matrix werden in der oberen Hälfte die verzerrten Generalized Scatter Plots und in der unteren Hälfte die ursprünglichen Streudiagramme verwendet.

ist. Zur besseren Veranschaulichung wird in Abbildung 3.14 ein Screenshot des Programms gezeigt.

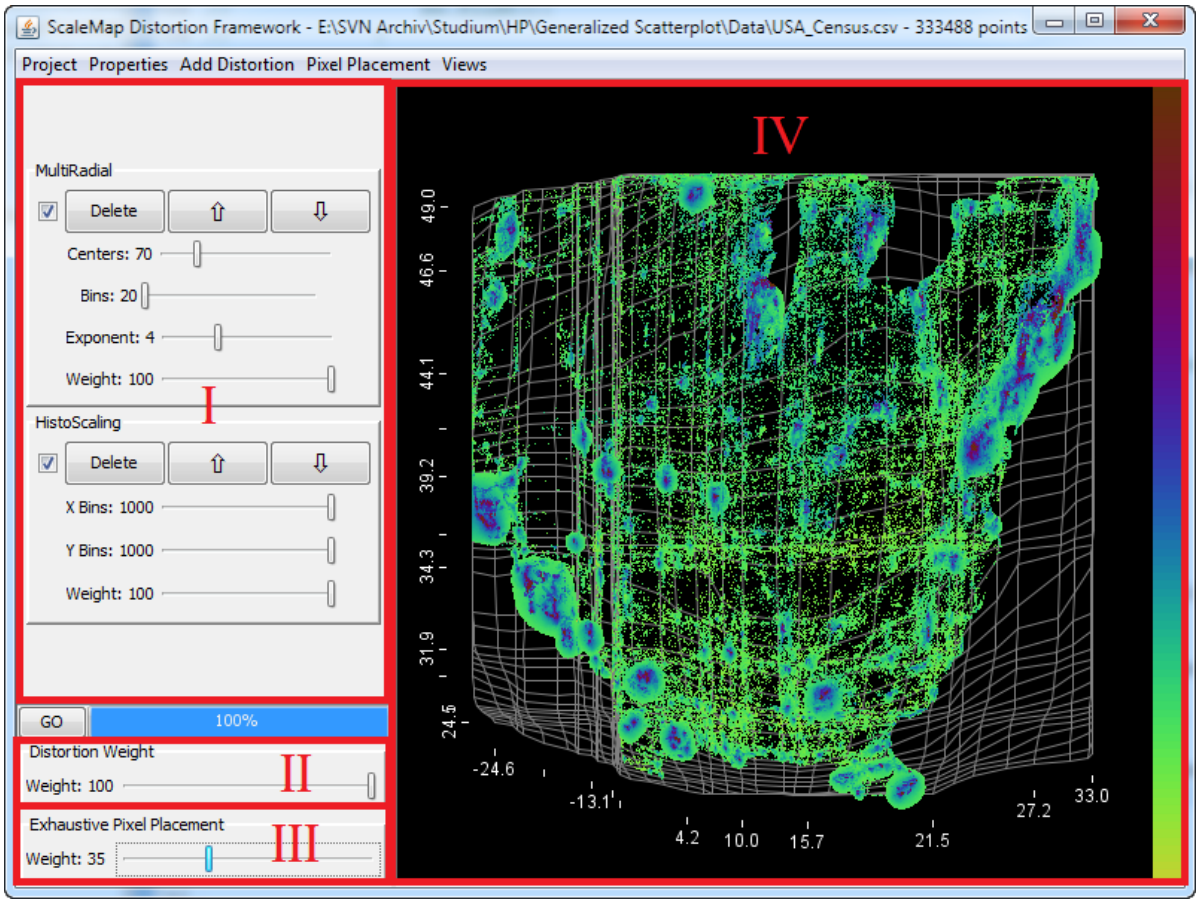



Abbildung 3.14: Dieser Screenshot zeigt die Referenzimplementierung bei der Visualisierung eines amerikanischen Zensusdatensatzes zur Einkommensverteilung. Der Bereich I auf der linken Seite bietet eine Übersicht der verwendeten Verzerrungstechniken und zusätzlich Einstellmöglichkeiten für die jeweiligen Parameter. Ferner kann mit II der Verzerrungsgrad aller Verzerrungen gemeinsam gesteuert werden und der Schieberegler in III steuert das Pixel Placement. Das Ergebnis kann auf der Zeichenfläche im Bereich IV mittels verschiedener Sichten auf die Daten begutachtet werden.

Der Bereich I in Abbildung 3.14 zeigt eine Übersicht der ausgewählten Verzerrungen. Die Verzerrungen können hierbei in beliebiger Reihenfolge und mit beliebigem Einfluss auf das Endergebnis aufgerufen werden. Anhand der Liste von ausgewählten Verzerrungen behält der Benutzer immer den Überblick und kann zusätzlich bei jeder einzelnen Verzerrung Parameter anpassen oder beispielsweise noch die Reihenfolge der Verzerrungen nachträglich ändern. Ferner kann mit II der Einfluss aller Verzerrungen gesteuert werden, wobei das Gewicht 0 für den ursprünglichen Datensatz steht. Der Schieberegler in III steuert, wie viele Datenpunkte gemäß dem Pixel Placement - Algorithmus versetzt werden. Welche Pixel Placement - Technik

verwendet werden soll, kann der Benutzer im Menü unter „Pixel Placement“ auswählen. Das Ergebnis der verzerrten und versetzten Datenpunkte kann auf der Zeichenfläche im Bereich IV begutachtet werden. Zur weiteren Datenanalyse stehen noch unterschiedliche Sichten auf das Ergebnis zur Verfügung, damit der Benutzer die Dichteverteilung besser abschätzen kann. Hierfür wurden beispielsweise sowohl Isolinien als auch Isoebenen implementiert.

Die Generalized Scatter Plots lösen die in der Motivation beschriebenen Probleme und bieten dem Benutzer zusätzlich einen hohen Grad an interaktivem Einfluss. Gerade die Berechnungen beim Exhaustive Pixel Placement zur Erzeugung beliebiger Zwischenstufen machen das Verfahren so interessant für die visuelle Datenanalyse. Erst dadurch, dass sich der eingestellte Parameter auf den Radius und nicht direkt auf den Grad an Überdeckung auswirkt und die angepasste Rundung auf ganze Zahlen, wurde eine lineare Relation zwischen Parameter und Ergebnis hergestellt. Ferner sind die vorgestellten Verfahren alle effizient und durch die Verwendung von heuristischem Pixel Placement auch sehr gut skalierbar. Für Zwischenergebnisse eignet sich nämlich das heuristische Pixel Placement, während für das Endergebnis das etwas länger dauernde Exhaustive Pixel Placement verwendet werden sollte. Dem Benutzer wird insgesamt also ein sehr mächtiges Werkzeug in die Hand gegeben, welches nicht nur viele Einstellmöglichkeiten zum Auffinden der optimalen Visualisierung bietet. Schließlich kann mittels der vorgegebenen Optimierungsfunktionen eine für die meisten Fälle zumindest ausreichende Startvisualisierung gefunden werden.

# 4 Fallstudien

Die im vorherigen Kapitel vorgestellten Generalized Scatter Plots zeichnen sich gerade durch den hohen Grad an Einflussnahme auf Verzerrung und Pixel Placement - Techniken aus. Dies resultiert in der völligen Freiheit des Benutzers, beliebige Einstellungen vorzunehmen, um die bestmögliche Sicht auf die Daten zu erhalten. Nachdem zuvor vor allem die technische Umsetzung im Vordergrund stand, soll nun anhand von Fallstudien geklärt werden, welche Vorteile die Generalized Scatter Plots bei der visuellen Datenanalyse haben. Hierzu wurden zwei Datensätze aus der realen Welt ausgesucht, welche als Datengrundlagen für die Generalized Scatter Plots dienen. Als erstes wird dabei ein Datensatz von Telefonkonferenzen mit 37 788 Einträgen untersucht. Danach kommt ein Zensusdatensatz der Vereingten Staaten von Amerika zum Zuge, welcher Aussagen über das Einkommen von Haushalten macht. Hierbei wurden aus Gründen des Datenschutzes mehrere Haushalte zusammengefasst und der Median der Einkommen berechnet. Diese Aggregation resultiert in einen Datensatz mit 333 488 Einträgen.

## 4.1 Nutzungsanalyse eines Telefonkonferenzsystems

Bei der Analyse eines Telefonkonferenzsystems ist für den Betreiber meist interessant, wie sich die Menge an Telefongesprächen im Datenraum verteilt. Zusätzlich sind auch die typische Konferenzdauer sowie die häufigsten Kosten wichtig, da dies beispielsweise Einfluss auf künftige Tarifabschlüsse haben kann. Ferner sollte auch die Korrelationsuntersuchung der Kosten in Relation zu der Anzahl der Konferenzteilnehmer möglich sein.

Der Datensatz, der hier untersucht werden soll, ist dreidimensional und besteht aus insgesamt 37 788 Einträgen. Er stammt aus dem Telefonkonferenzsystem, welches Hewlett Packard für nationale und internationale Telefonkonferenzen verwendet. Die erste Dimension beschreibt dabei die Dauer einer Telefonkonferenz in Minuten. Die zweite Dimension befasst sich hingegen mit den Kosten einer jeden Konferenzschaltung in US Dollar. Zu guter Letzt wird in der dritten Dimension die jeweilige Anzahl von Teilnehmern der Telefonkonferenz festgehalten. Da die meisten Konferenzen weniger als 100 Teilnehmer hatten und nur drei mehr als 500 – der Maximalwert liegt bei 2000 Teilnehmern – bot sich eine logarithmische Farbskalierung der Teilnehmerzahlen an. Daher wird in allen folgenden Abbildungen, bei denen die Farbe die Teilnehmerzahl repräsentiert, eine logarithmische Skalierung verwendet. Auch wenn bei allen Visualisierungen dieses Datensatzes der Anschein entsteht, es seien (absichtlich) ungünstige Maxima für die x- und die y-Achse gewählt worden, so trügt dies. Es befinden sich nämlich noch Datenpunkte in der oberen rechten Ecke, welche zu diesem für traditionelle Streudiagramme schlecht visualisierbaren Datenraum führt.

Wie schon erwähnt wurde, kann der Verzerrungsgrad und die Menge an Pixel Placement

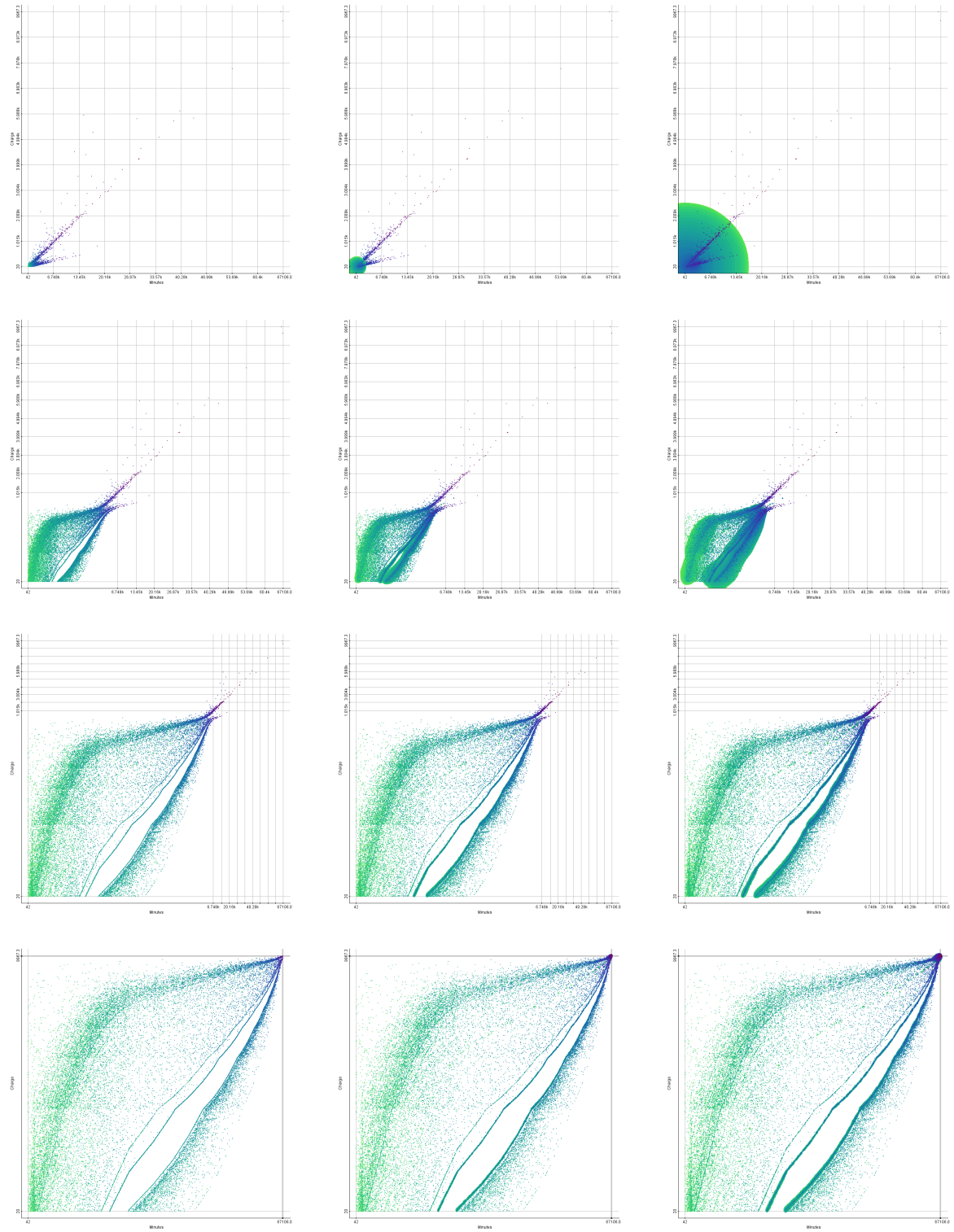


Abbildung 4.1: Diese Grafiken zeigen verschiedene Generalized Scatter Plots für Telefonkonferenznutzungsdaten. Auf der x-Achse wird die Anrufdauer und auf der y-Achse werden die Gesprächskosten abgetragen. Die Farbe zeigt die Anzahl der Teilnehmer (von grün für wenige bis hin zu braun für viele Teilnehmer). Das ursprüngliche Streudiagramm wird oben links gezeigt, die Überdeckung wurde schrittweise reduziert (von links nach rechts) und die Verzerrung schrittweise verstärkt (von oben nach unten).

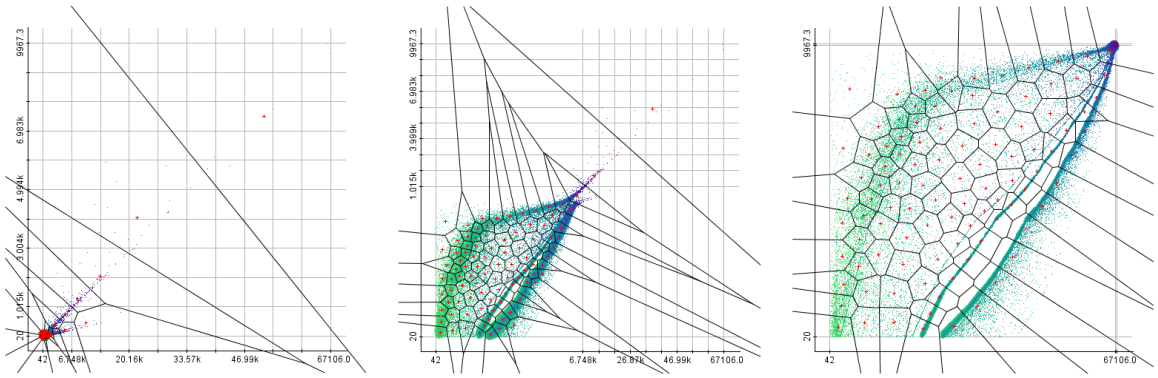
unabhängig voneinander kontrolliert werden. Die somit beliebig kombinierbaren Einstellungen können interaktiv untersucht werden. Um das Spektrum der möglichen Ansichten aufzuzeigen, wurden auf der vorherigen Seite in Abbildung 4.1 zwölf beispielhafte Ergebnisse herausgegriffen. Dabei werden jeweils verschiedene Generalized Scatter Plots für verschiedene Verzerrungsgrade und Mengen an Pixel Placement gezeigt. Die Grafik oben links zeigt ein normales Streudiagramm ohne Verzerrung und maximale durch die Daten erzeugte Überdeckung. Von links nach rechts wird die Überdeckung zuerst auf 50 Prozent und anschließend auf 0 Prozent reduziert. Von oben nach unten wird der Verzerrungsgrad schrittweise auf 28 Prozent, 60 Prozent und zuletzt auf 100 Prozent erhöht. Die Ansicht mit der stärksten Verzerrung und ohne Punktüberdeckungen findet sich somit unten rechts.

Beim Vergleich der verschiedenen Varianten fällt auf, dass das ursprüngliche Streudiagramm nur sehr wenig Information enthüllt. Man erkennt gerade einmal zwei unterschiedliche lineare Zusammenhänge, und ferner ist es sehr schwierig, die Gesamtzahl an Telefonkonferenzen auszumachen. Ferner ist der Zusammenhang zwischen Teilnehmeranzahl, Kosten und Gesprächsdauer auf Grund des hohen Überdeckungsgrads nur schwer zu erkennen. Mit zunehmender Verzerrung (von oben nach unten) und zunehmendem Pixel Placement (von links nach rechts) werden die hochgradig geclusterten Daten partitioniert und es werden mehr Details sichtbar. Es werden bei mittlerer Verzerrung schon mindestens zwei verschiedene lineare Zusammenhänge ersichtlich, welche sich bei noch stärkerer Verzerrung in mindestens vier unterschiedliche Kurven unterteilen. Letztendlich werden bei minimaler Überdeckung und vollständiger Verzerrung interessante Details sichtbar, welche weder im ursprünglichen Streudiagramm noch bei den meisten anderen Techniken zu sehen waren. In dieser finalen Ansicht können bis zu neun unterschiedliche Kurven ausgemacht werden, welche jeweils einem bestimmten Tarif entsprechen.

Bei der Analyse des Datensatzes mit den Generalized Scatter Plots konnten die folgenden Korrelationen zwischen Gesprächsdauer, Kosten und Teilnehmeranzahl beobachtet werden. Die linke Kurve zeigt, dass Telefonate, die vom Verhältnis von Gesprächsdauer zu Kosten die teuersten sind, die meisten Datenpunkte stellen. Wobei es jedoch eine hohe Streuung innerhalb der Kosten gibt. Interessanterweise sind die, vom Tarif her gesehen, teuersten Gespräche nationale Gespräche. Zusätzlich gibt es eine mittlere Kurve, welche signifikant billiger ist, jedoch eine offensichtliche Korrelation zwischen Kosten und Sekunden zeigt. Diese Kurve ist das Ergebnis eines Spezialtarifs nach Kanada, welche nur für eine geringe Anzahl von Teilnehmern verwendet werden kann (sie besteht nur aus grünen Punkten). Der rechte Bereich beinhaltet vor allem internationale Gespräche. Die drei blauen Linien repräsentieren dabei drei verschiedene Dienstleister (AT&T, Sprint und ConCall). Die rechte (dickste) Kurve (AT&T) verzeichnet die meisten Telefongespräche und besitzt einen hohen Grad an Überdeckung von Punkten. Die Dicke der Kurven offenbart die Menge an nationalen und internationalen Gesprächen, wobei der Vergleich der Dicke zu einem weiteren interessanten Detail führt. Die internationalen Gespräche haben bei jedem Anbieter eine einfachere Tarifstruktur (durchgezogene Linien) während die nationalen Gespräche variabler sind und von weiteren Einflussfaktoren abhängen. Diese wurden jedoch nicht in der Visualisierung gezeigt. Hierzu zählen beispielsweise Tageszeit oder, ob der Tag ein Werktag bzw. Feiertag ist.

Zur visuellen Datenanalyse gehören natürlich auch Data Mining - Techniken, um die Daten weitergehend zu untersuchen. Beispielsweise sollen signifikante Gruppierungen beziehungs-

weise Muster in den Daten aufgespürt werden, in dem Clustering - Techniken angewendet werden. Zur einfachen Überprüfung des Konzepts wurde der k-Means - Algorithmus [20] implementiert und das Ergebnis mittels Voronoizellen visualisiert. Diese sehr einfachen Verfahren können der jeweiligen Datengrundlage angepasst werden, da sie modular austauschbar sind. Der k-Means - Algorithmus unterteilt die Daten in $k$ Cluster, wobei jeder Punkt demjenigen Cluster zugesprochen wird, bei dem der Punkt zum Clusterzentrum den geringsten Abstand hat. Die hieraus resultierenden Cluster sollten also zur Datenverteilung korreliert sein. Die Voronoi - Tessellierung wird anschließend auf die vom k-Means - Algorithmus gefundenen Clusterzentren angewendet, um visuell aufzuzeigen, zu welchem Cluster die Punkte jeweils gehören. Kleine rote Markierungen wurden dabei verwendet, um die Clusterzentren zu repräsentieren, und die Grenzen der Voronoizellen wurden mit schwarzen Linien eingezeichnet.



(a) keine Verzerrung (b) 50 Prozent Verzerrung (c) 100 Prozent Verzerrung

Abbildung 4.2: Auswirkungen des Verrungsgrades auf die visuelle Datenanalyse mit k-Means ($k = 100$). Voronoizellen wurden zur visuellen Repräsentation der Cluster und rote Markierungen für die Clusterzentren verwendet.

In Abbildung 4.2 wird das Ergebnis der oben beschriebenen Verfahren für verschiedene Verzerrungsgrade gezeigt. Von links nach rechts steigt der Verzerrungsgrad von null auf fünfzig bis hin zu hundert Prozent. Aus den Abbildungen geht deutlich hervor, dass eine zunehmende Verzerrung die Ergebnisse des Clusterings durchaus verbessern kann. Der ursprüngliche Datensatz (Abbildung 4.2(a)) wird sehr schlecht partitioniert und das Ergebnis ist eine Ansammlung von Clusterzentren, welche die Daten kaum vernünftig segmentieren kann. Durch den mittleren Verzerrungsgrad in Abbildung 4.2(b) können schon mehr Details und mehr unterscheidbare Segmente ausfindig gemacht werden. Schlussendlich enthüllt die maximale Verzerrung in Abbildung 4.2(c) die vorher in den Daten versteckten Muster. Die Clusterzentren folgen den Linien mit hoher Dichte und verbessern die Sichtbarkeit dieser Muster. Zusätzlich unterteilen die Voronoizellen die Datenpunkte entlang der aufgefundenen Muster.

## 4.2 Zensusdaten der Vereinigten Staaten von Amerika

Der zweite Datensatz, der untersucht werden soll, stammt aus einem amerikanischen Zensus aus dem Jahre 1999 [21]. Dieser befasst sich mit der geographischen Einkommensverteilung in den Vereingten Staaten von Amerika. Hierbei wurden aus Gründen des Datenschutzes mehrere Haushalte zusammengefasst und der Median der aggregierten Einkommen berechnet. Der so gewonnene Datensatz umfasst 333 488 Einträge, die aus einer geographischen Position und dem jeweiligen Einkommen bestehen. Anhand dieser Daten soll gezeigt werden, welchen großen Vorteil die Integration verschiedener Verzerrungstechniken in ein einzelnes Programm birgt. Schließlich ermöglichen die Generalized Scatter Plots gerade die Auswahl der anzuwendenden Verzerrungen und unterstützen auf diese Weise die visuelle Datenanalyse. Die Ungleichverteilung der Daten wird in den nachfolgenden Abschnitten von verschiedenen Verzerrungstechniken behandelt. Wobei diesmal das Pixel Placement nicht im Vordergrund steht, daher wurde in allen Abbildungen der Verzerrungsergebnisse 75 Prozent Exhaustive Pixel Placement angewendet.

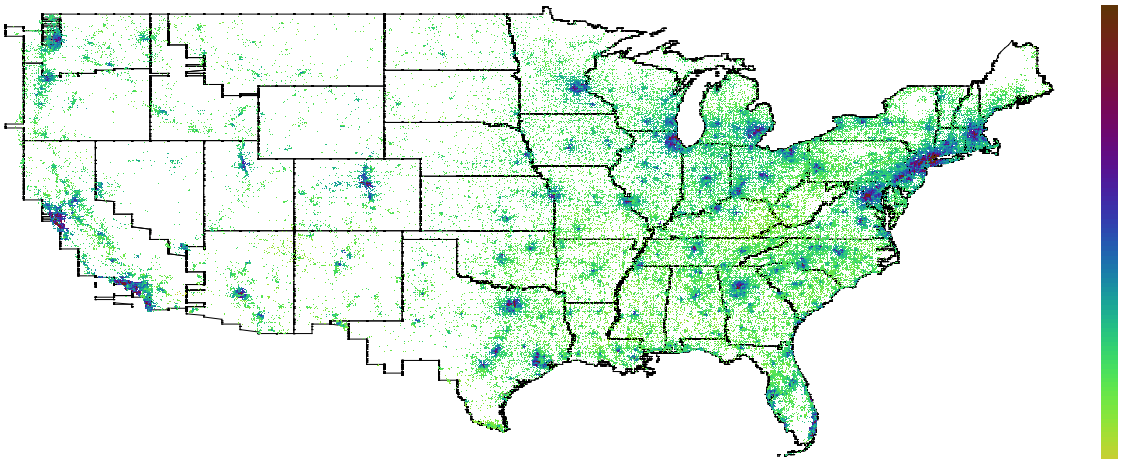


Abbildung 4.3: Ein traditionelles Streudiagramm (ohne Verzerrung und ohne Pixel Placement) mit den Polygonzügen der Vereinigten Staaten von Amerika. Jeder Punkt steht für eine aggregierte Menge von Haushalten, wobei die Farbe den Median der Einkommen angibt. Der verwendete Colormap geht hierbei von grün für niedrige Einkommen über blau bis hin zu braun für hohe Einkommen.

Das traditionelle Streudiagramm zur Visualisierung des Datensatzes wird in Abbildung 4.3 verwendet, wobei zusätzlich die Polygonzüge der Staatengrenzen der Vereinigten Staaten von Amerika eingezeichnet wurden. Jeder Punkt steht für eine aggregierte Menge von Haushalten, wobei die Farbe den Median der Einkommen angibt. Der verwendete Colormap geht hierbei von grün für niedrige Einkommen über blau bis hin zu braun für hohe Einkommen. Auch schon im ursprünglichen Streudiagramm ist eine deutliche Ungleichverteilung der Haushalte erkennbar. Beispielsweise sind die Punkte im Mittleren Westen der Vereinigten Staaten von Amerika rar gesät, während im Osten deutlich mehr Haushalte liegen.

In Abbildung 4.4 wurden die Daten durch die achsenparallelen HistoScale - Technik verzerrt. Deutlich sichtbar ist eine Vergrößerung der Oststaaten und eine sehr starke Verkleine-

rung der Staaten im Mittleren Westen. Anhand dieser Grafik werden auch die Schwächen des HistoScale - Verfahrens aufgedeckt. Beispielsweise wird im Westen der Staat Nevada sehr stark vergrößert, obwohl er nur sehr wenige Haushalte enthält. Dies liegt daran, dass Nevade im selben Längengradbereich der bevölkerungsreichen Region um Los Angeles liegt und zudem noch im selben Breitengradbereich wie die bevölkerungsreichen Regionen an der Ostküste.

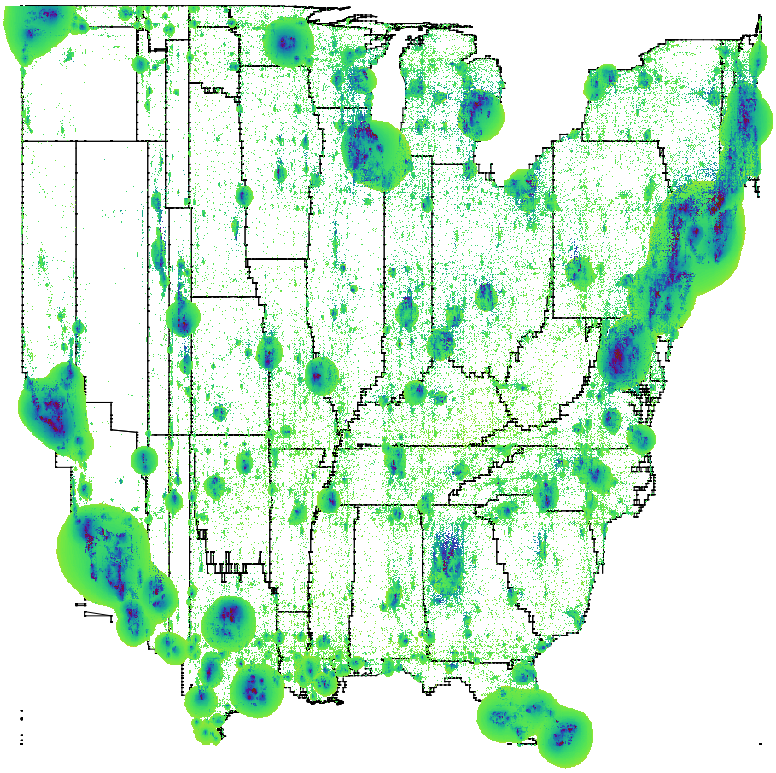


Abbildung 4.4: Verzerrung des Zensusdatensatzes mit der HistoScale - Technik und 75 Prozent Pixel Placement.

Allem Anschein nach ist die HistoScale - Technik somit für diesen Datensatz nicht optimal geeignet. Daher bietet es sich an, eine andere Verzerrungstechnik auf diesen Datensatz anzuwenden. In Abbildung 4.5 wurde zunächst die MultiAngular - Technik ausgewählt. Das MultiAngular - Verfahren führt dabei zu einer sehr starken Verzerrung der Vereinigten Staaten von Amerika, da alle Staaten um den Mittleren Westen herum gedreht wurden. Dies hat zum einen den Vorteil, dass im äußeren Bereich mehr Platz zur Verfügung steht, aber zum anderen den Nachteil, dass der Mensch nicht gewohnt ist, die USA in einer kreisförmigen Struktur zu sehen. Zudem wird der – von der Punktedichte her – uninteressante Mittlere Westen ins Zentrum der Visualisierung und damit in den Mittelpunkt der Aufmerksamkeit gerückt. Ferner ist es bei diesem Ergebnis letztendlich sehr schwer, die ursprüngliche geographische Lage der Datenpunkte zu rekonstruieren. Gerade bei geographischen Daten muss die ursprüngliche Orientierung anhand der Himmelsrichtungen beibehalten werden, um ein Verständnis der Daten zu fördern.

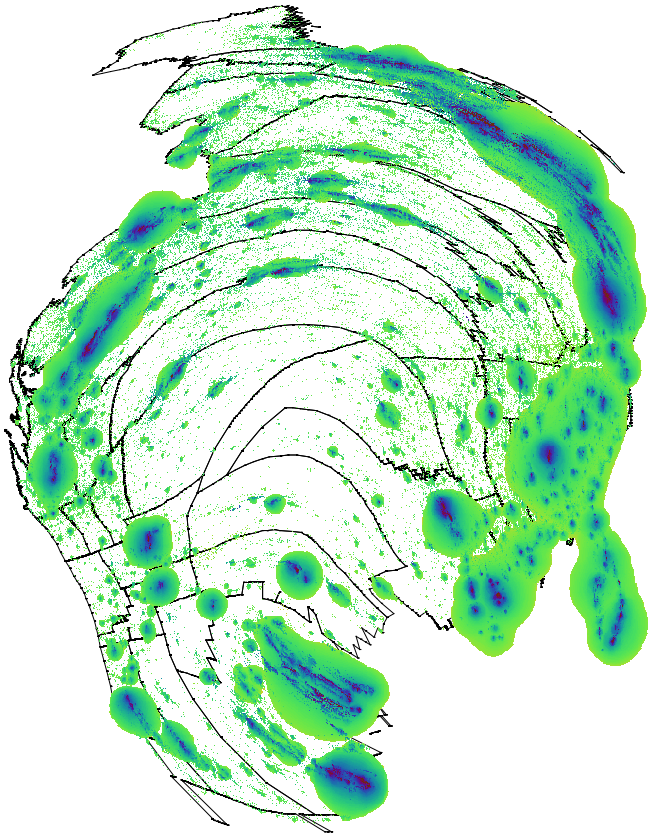


Abbildung 4.5: Verwendung der MultiAngular - Technik und 75 Prozent Pixel Placement.

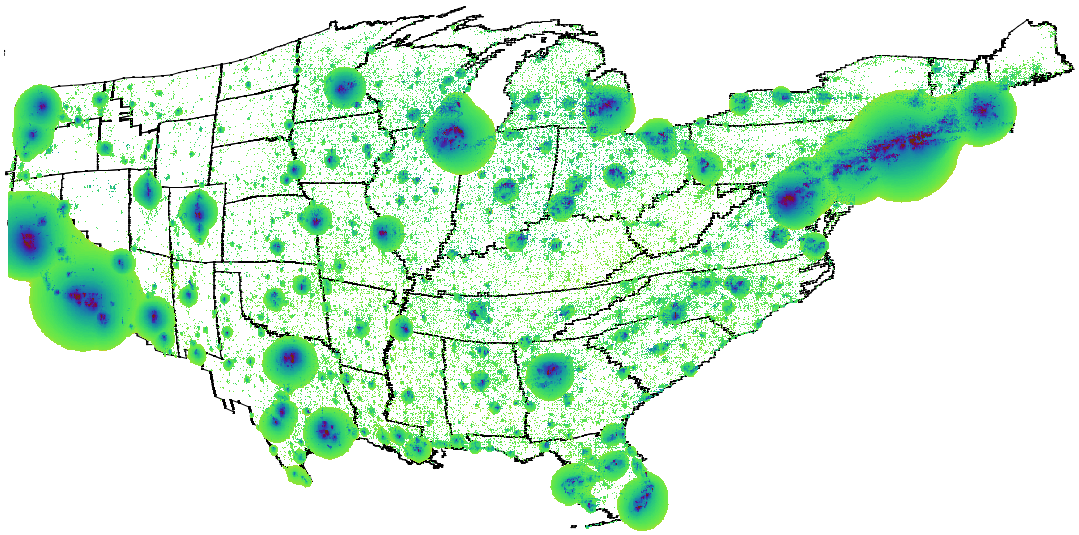


Abbildung 4.6: MultiRadial - Verzerrung und 75 Prozent Pixel Placement.

Nachdem sich nun die MultiAngular - Technik als noch schlechter als das HistoScale - Verfahren erwiesen hat, kann nun noch als Letztes der MultiRadial - Ansatz ausgewählt werden. Das Ergebnis dieser Technik kann in Abbildung 4.6 begutachtet werden. In diesem Fall ist die Grundform der Vereinigten Staaten trotz Verzerrung noch gut erkennbar. Wie erwartet wird der bevölkerungsschwache Mittlere Westen verkleinert, während bevölkerungsstarke Regionen vergrößert werden. Gleichzeitig bleiben aber Daten in der selben Orientierung wie im traditionellen Streudiagramm, was die visuelle Datenanalyse deutlich erleichtert. Anscheinend liefert der MultiRadial - Ansatz somit die beste Verzerrung dieses Datensatzes.

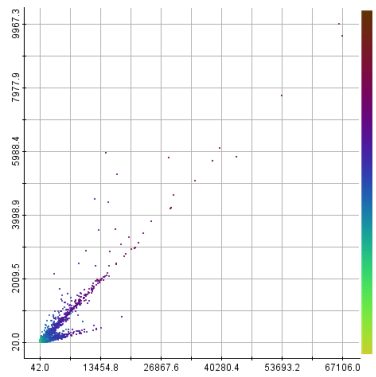
Wie an diesem Fallbeispiel deutlich geworden ist, haben die Generalized Scatter Plots den großen Vorteil, mehrere verschiedene Techniken in einer einzigen Applikation zu integrieren. Zudem wurde in dem vorangegangenen Fallbeispiel deutlich, dass die Interaktivität und die freie Einflussnahme des Benutzers auf einzelne Einflussfaktoren zu den Stärken der Generalized Scatter Plots gehört. Selbstverständlich ist der hier vorgestellte Ansatz ein Programm, welches sich vor allem an Experten auf dem Gebiet der visuellen Datenanalyse richtet. Aber gerade von den vielfältigen Möglichkeiten, welche dieser Ansatz bietet, kann der Benutzer profitieren. So sind durchaus Anwendungsfälle denkbar, bei denen erst eine gewichtete Kombination von Verzerrungstechniken zur bestmöglichen Ansicht auf die Daten führt.

# 5 Evaluierung

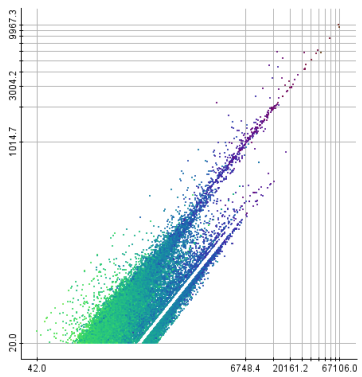
Jedes neu entwickelte Verfahren kann noch so gut konzeptionell erdacht sein, aber erst durch den Vergleich mit schon existierenden Ansätzen werden Vorzüge und Nachteile besonders gut ersichtlich. Aus diesem Grund werden in diesem Kapitel die Generalized Scatter Plots mit einigen ausgewählten schon existierenden Techniken verglichen. Dabei bietet es sich an, die Datensätze aus dem vorherigen Kapitel zu verwenden, weil die entsprechenden Generalized Scatter Plots schon abgebildet und beschrieben wurden. Die Fragestellung bei dieser visuellen Evaluierung lautet, wie ersichtlich die in den Generalized Scatter Plots entdeckten Muster sind. Nachdem gezeigt wurde, dass einzelne angewendete Techniken gerade beim Telefondatensatz ihre Grenzen erreichen, werden die Generalized Scatter Plots mit Kombinationen bestehender Techniken verglichen.

Zunächst werden für einen visuellen Vergleich in Abbildung 5.1 sieben verschiedene Verfahren zur Visualisierung des Telefondatensatzes verwendet. Beim herkömmlichen Streudiagramm (a) ist die Punktedichte kaum abzusehen und abgesehen von globalen Phänomenen sind detailliertere Analysen nur schwer möglich. Eine weitere übliche Technik, welche angepasst an die Daten bessere Visualisierungen erzeugt, ist die Achsenskalierung. Hierbei sind sowohl Quadratwurzel- also auch logarithmische Skalierungen üblich. Die letztere Technik kann in (b) begutachtet werden. Auch werden Daten manchmal künstlich verrauscht, um Punktüberdeckungen zu reduzieren und gleichzeitig die Dichte zu visualisieren. In (c) wurde den Daten absichtlich leichtes Rauschen hinzugefügt, um genau dies zu erreichen. Allerdings kann auch ein Verrauschen der Daten den hohen Grad an Punktüberdeckung nicht kompensieren.

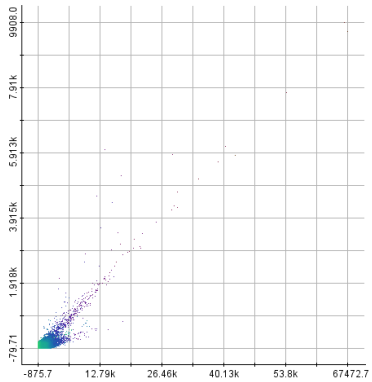
Nach den eher statistischen Variationen zur Visualisierung des Datensatzes, werden nun typische Visualisierungstechniken verwendet. So sind Interaktionsmöglichkeiten wie beispielsweise Zooming und Panning bei der visuellen Datenanalyse weit verbreitet. Daher wurde in der Abbildung (d) ein Ausschnitt des Datenraums vergrößert, wobei allerdings die Gesamtübersicht über die Daten schnell verloren geht. Zusätzlich ist das Verfahren zur Zeichnung halb transparenter Punkte häufig anzutreffen. Durch die Überzeichnung von überdeckenden Datenpunkten werden dichtere Bereiche undurchsichtig und nicht so dichte Bereiche bleiben durchscheinend. Dieser Ansatz wurde in (e) verfolgt und liefert nur eine grobe Dichteabschätzung für den Datensatz. Außerdem besteht auch die Möglichkeit, den Datenraum zu unterteilen und eine Aggregation bezüglich der Punktanzahl durchzuführen. Das HexBin - Verfahren unterteilt hierzu den Datenraum mittels Hexagonen und bestimmt für jeden Bereich die Punktdichte. Diese wird mittels Graustufen anschließend in den Datenraum abgetragen, wobei das Ergebnis in (f) zu sehen ist. Deutlich sichtbar ist eine hohe Dichtekonzentration in der linken unteren Ecke, während sich – relativ gesehen – sehr wenige Datenpunkte im übrigen Bereich des Datenraums befinden. In Abbildung (g) wurde die Sampling - Technik auf den Telefondatensatz angewendet. Obwohl nur vierzig Prozent der Daten dargestellt wurden, bleibt eine hohe Punktüberdeckung im unteren linken Bereich.



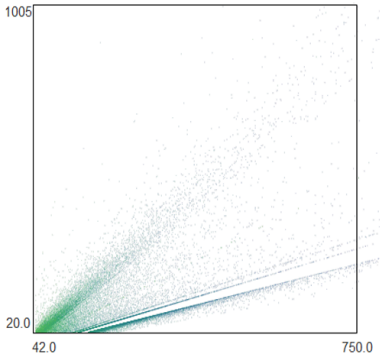

(a) Traditionelles Streudiagramm



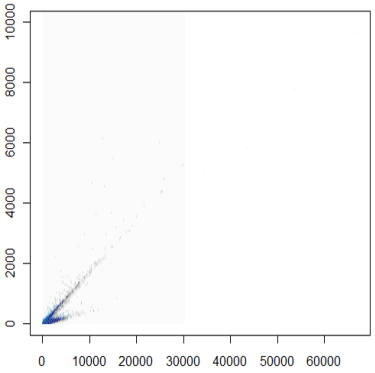

(b) Logarithmische Achsenskalen



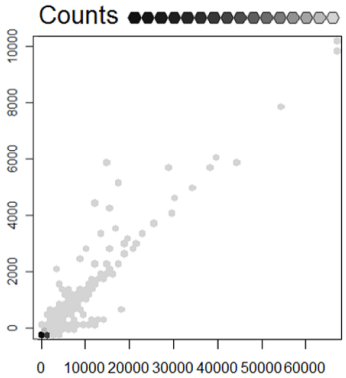

(c) Künstliches Verrauschen



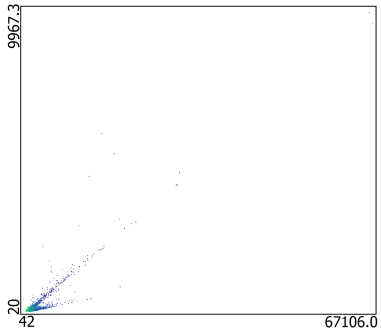

(d) Ausschnittvergrößerung




(e) Alpha Blending



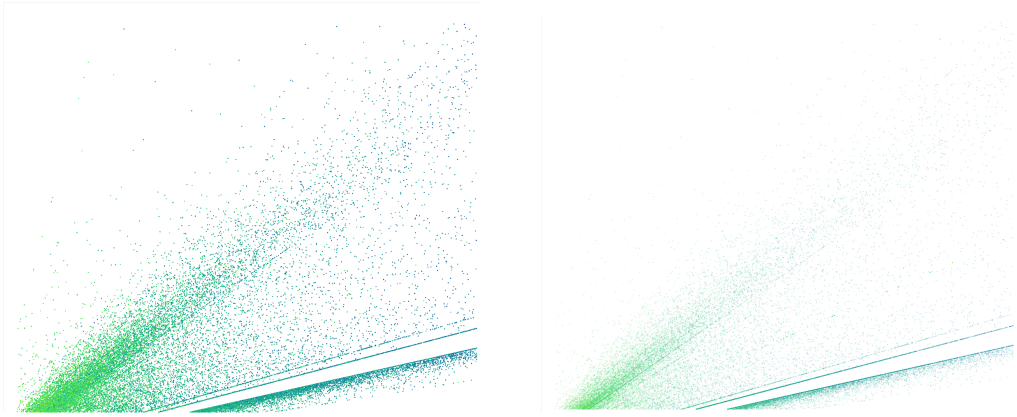

(f) HexBin - Technik




(g) Sampling

Abbildung 5.1: Herkömmliche Ansätze zur Lösung des Überdeckungsproblems zeigen bei diesem Datensatz ihre Grenzen. Verwendet wurde der Telefondatensatz, wobei die x-Achse die Gesprächsdauer und die y-Achse die jeweiligen Telefonkosten beschreiben. Bei den farbigen Abbildungen konnte noch die dritte Dimension (Teilnehmeranzahl) logarithmisch skaliert dargestellt werden.

Nachdem in Abbildung 5.1 jeweils einzelne Techniken zur Visualisierung des Datensatzes verwendet wurden, soll nun eine Kombination von zwei Techniken ausprobiert werden. In der Abbildung 5.2 wurden zusätzlich zur Ausschnittsvergrößerung die Datenpunkte halbtransparent eingezeichnet. Zum besseren Vergleich wurde die alleinige Vergrößerung neben der kombinierten Variante abgebildet. Erst durch die Kombination beider Techniken werden die in Abbildung 4.1 gefundenen Muster in den Daten sichtbar. Nachteil der Ausschnittsvergrößerung ist allerdings, dass nicht mehr alle Daten gleichzeitig dargestellt werden und die Gesamtübersicht verloren geht.



(a) Ausschnittsvergrößerung

(b) Ausschnittsvergrößerung und Alpha Blending

Abbildung 5.2: Erst die Kombination mehrerer Techniken lässt die durch die Generalized Scatter Plots gefundenen Muster sichtbar werden. Zur besseren Veranschaulichung der Auswirkungen von Alpha Blending zur Dichtedarstellung wurde zunächst nur eine Ausschnittsvergrößerung vorgenommen (links) und die Punkte anschließend halbtransparent gezeichnet (rechts).

Zusätzlich zum Telefondatensatz wird nun auch anhand des USA-Zensus gezeigt, dass eine Kombination der Techniken ähnlich gute Ergebnisse bringt. In diesem Fall wurde ein fünfprozentiges Sampling durchgeführt und die Punkte wurden zusätzlich semitransparent gezeichnet. Die Semitransparenz ermöglicht hierbei die Dichteabschätzung, um die geografische Verteilung und Häufung von Haushalten besser abzuschätzen. Durch das Sampling wird die Überdeckung von Punkten gemindert und zudem eine visuelle Überladung der Visualisierung verhindert. In Abbildung 5.3 wird das Ergebnis der Kombination dieser zwei Techniken gezeigt. Gut erkennbar sind dicht besiedelte Bereiche an der Ost- und Westküste.

Bei genauer Betrachtung der Generalized Scatter Plots für den amerikanischen Zensusdatensatz fällt ein typisches, geografisches Muster auf. Typischerweise wohnen inmitten der Stadt nicht die reichen Mennschen, sondern eher etwas außerhalb. Direkt im Zentrum der Stadt findet man typischerweise ärmere Haushalte. Am Beispiel von Chicago kann man dies in Abbildung 5.4 begutachten. Zum Vergleich wurde derselbe Bereich in der kombinierten Technik auch vergrößert. Hierbei wird deutlich, dass das Pixel Placement und besonders die Reihenfolge des Pixel Placements großen Einfluss auf die Sichtbarkeit dieses Phänomens haben. Beim

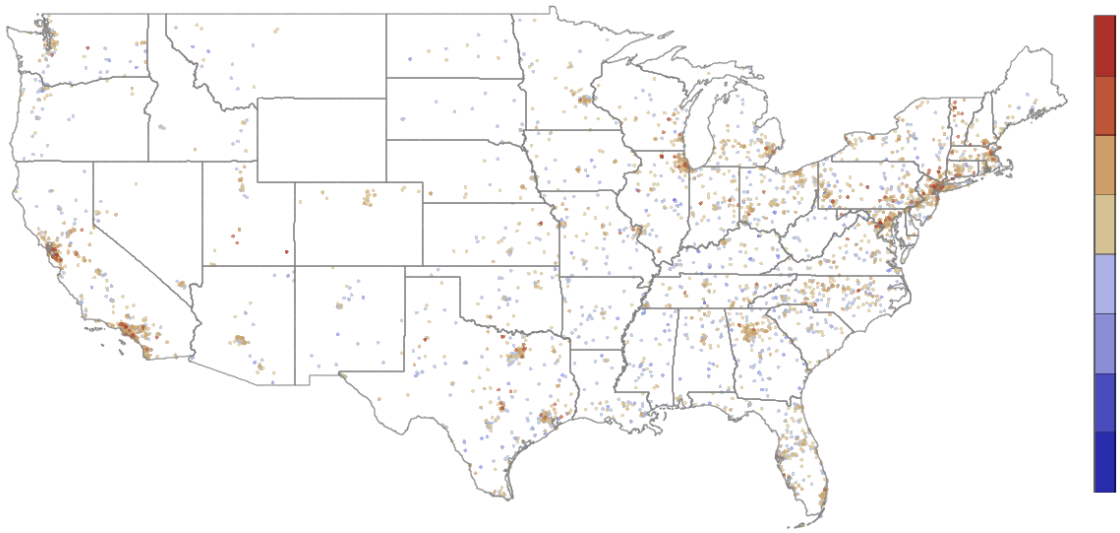
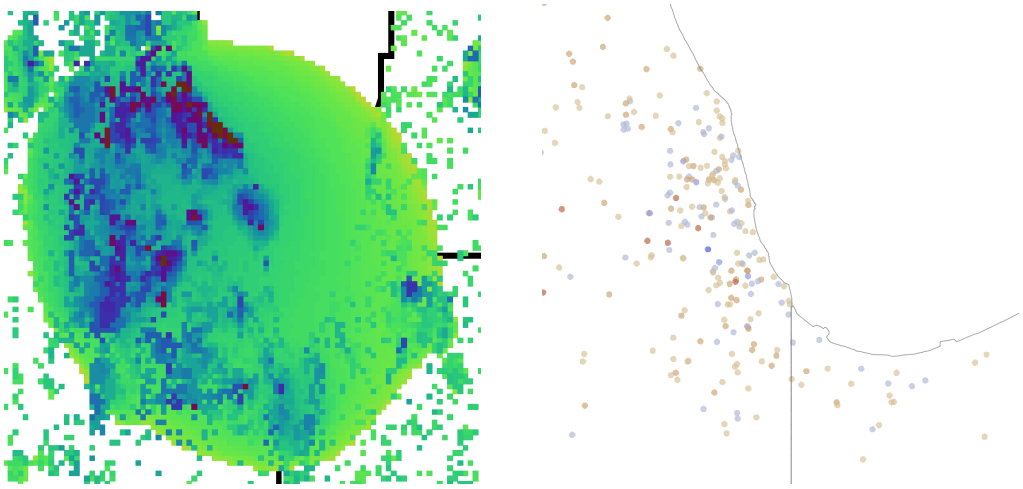


Abbildung 5.3: Visualisierung der amerikanischen Zensusdaten unter Verwendung von Sampling und Alpha Blending. Der verwendete Colormap repräsentiert das Medianeinkommen mit blauer Farbe für niedrige Einkommen bis hin zu rot für hohe Einkommen.



(a) Ausschnitt aus der Abbildung 4.4

(b) Ausschnitt aus der Kombination von Sampling und Alpha Blending

Abbildung 5.4: Typische Einkommensverteilung in einer Stadt am Beispiel von Chicago: Direkt im Zentrum wohnen ärmere Menschen und etwas außerhalb die reicheren. Dieses Phänomen wird durch die Reihenfolge des Pixel Placements besonders gut sichtbar gemacht. Beim Sampling mit Alpha Blending, welches für die Gesamtansicht der Vereinigten Staaten sehr gut war, ist dieses Phänomen nicht mehr ersichtlich.

Sampling mit Alpha Blending, welches für die Gesamtansicht der Vereinigten Staaten sehr gut war, ist dieses Phänomen nicht mehr ersichtlich.

Die Generalized Scatter Plots können beim Auffinden von Mustern helfen und den Benutzer auf interessante Zusammenhänge stoßen, die ihm beim Betrachten eines einfachen Streudiagramms entgehen würden. Zusätzlich haben die Generalized Scatter Plots den Vorteil, dass sie sehr ähnlich zum ursprünglichen Streudiagramm sind – in der Ansicht mit keiner Verzerrung und keinem Pixel Placement entsprechen sie vollkommen dem traditionellen Streudiagramm. Anders als beispielsweise beim HexBin - Ansatz müssen die Daten nicht künstlich aggregiert werden, was auch dazu beiträgt, eine möglichst gewohnte Ansicht auf die Daten zu gewährleisten. Dadurch werden auch die lokalen Nachbarschaftsbeziehungen nicht zerstört, was einer visuellen Datenanalyse sehr zuträglich ist. Beim Vergleich mit den Kombinationen von bestehenden Techniken fällt auf, dass die Generalized Scatter Plots ebenbürtig zu den Kombinationen mehrerer Techniken sind. Der Vorteil für den Benutzer liegt darin, dass er nicht die Kombination selber bestimmen muss, sondern einfach die Generalized Scatter Plots verwenden kann. Zudem bleibt durch das Pixel Placement ein hoher Detailgrad erhalten, welcher beispielsweise die typische geographische Einkommensverteilung in Städten sichtbar macht.

# 6 Diskussion und Ausblick

In dieser Arbeit wurden die Generalized Scatter Plots vorgestellt, eine neue Technik zur Visualisierung großer Mengen niederdimensionaler Daten. Der Ansatz ist eine Erweiterung der normalen Streudiagramme und befasst sich mit der Lösung des Überdeckungsproblems von Datenpunkten. Dabei werden die Datenpunkte mittels eines Pixels auf dem Bildschirm repräsentiert, wobei Verzerrungs- und Pixel Placement -Techniken verwendet werden, um die Punktüberdeckung zu verhindern.

Besonderes Augenmerk wurde auf die interaktive Einflussnahme des Benutzers auf den Verzerrungsgrad und die Menge an erlaubter Punktüberdeckung gerichtet. Der Benutzer kann völlig frei jegliche Zwischenstadien zwischen dem traditionellen Streudiagramm und der vollständig verzerrten Ansicht ohne jegliche Punktüberdeckung erzeugen. Dies ermöglicht in jedem Anwendungsgebiet, die bestmögliche Sicht auf die Daten zu generieren. Dabei mussten die Pixel Placement - Techniken entsprechend der Anforderung der interaktiven Einflussnahme angepasst und erweitert werden. Gerade beim Exhaustive Pixel Placement ist eine lineare Interpolation zwischen vollständig angewandter Technik und dem Ursprungszustand nicht einsetzbar, da visuelle Artefakte in Form von leeren Pixeln enstehen. Die Generalized Scatter Plots sind eine effiziente und effektive Lösung des Überdeckungsproblems und erleichtern die Exploration großer Datenmengen. Anhand von verschiedenen Datensätzen aus der realen Welt war es möglich, das breite Anwendungsspektrum der hier vorgestellten Technik aufzuzeigen. Dabei wurde durch die Ergebnisse ersichtlich, dass die Generalized Scatter Plots signifikant mehr Informationen enthüllen als die traditionellen Streudiagramme.

Zusätzlich wurde mittels Fehlerfunktionen für Überdeckung und Punktversetzung gezeigt, welche Auswirkungen Verzerrung und Pixel Placement haben. Hierdurch konnte ein optimaler Kompromiss zwischen Überdeckung und Verzerrung gefunden werden. Diese optimale Ansicht soll zukünftig als Einstiegsvisualisierung nach dem Laden der Daten verwendet werden, um den Benutzer bei der visuellen Datenanalyse zu unterstützen.

Ein zusätzlicher Vorteil liegt in der modularen Ansicht auf die Wirkungsweise von Verzerrungen. Durch die beliebige Auswahl, Reihenfolge und Stärke der Verzerrungen eröffnen sich weitere Benutzungsperspektiven. Zudem können automatische Datenanalysen den Benutzer unterstützen. Beispielsweise zeigen schon der k-Means - Algorithmus und die Voronoi - Tessellierung, dass sich die Verzerrung auch auf die Clustering - Ergebnisse positiv auswirken kann. Ferner können durch die Visualisierung in angepassten Streudiagramm - Matrizen mehrdimensionale Daten dargestellt werden, und es kann gleichzeitig ein Bezug zwischen verzerrter und traditioneller Ansicht hergestellt werden.

Selbstverständlich haben die Generalized Scatter Plots auch durchaus Nachteile, auf welche dieser Abschnitt nun gesondert eingehen wird. Grundsätzlich limitiert die Bildschirmauflösung die Möglichkeiten, Punkte überdeckungsfrei darzustellen, wenn Datenpunkte mittels Pixeln repräsentiert werden. Daher ist besonders das Pixel Placement - Verfahren davon be-

troffen, wenn der zur Verfügung stehende Platz nicht ausreicht. Anders als beispielsweise das Sampling - Verfahren, welches aktiv Rauschen verringert, gehen die Generalized Scatter Plots von nicht verrauschten Daten aus. Daher bieten die Generalized Scatter Plots auch keine Unterstützung für die Behandlung von Rauschen. Zudem arbeiten sie immer mit dem vollen Datensatz und führen keine Reduzierung der Daten durch, was automatisch zu Speicher- und Darstellungsproblemen führt. Ferner ist gerade die Anwendung von Verzerrungstechniken auf geographische Daten problematisch, weil die wichtigste Eigenschaft geographischer Daten gerade die räumliche Lage ist. Außerdem führt die Verzerrung dazu, dass räumliches Clustering im verzerrten Datenraum nicht mehr möglich ist. Zusätzlich unterstützen die Generalized Scatter Plots keine Regressionsanalyse, da die Regression sehr stark vom Pixel Placement beeinflusst werden kann. Auch wenn eigentlich ein kreisförmiges Pixel Placement durchgeführt wird, so ist es abhängig von der jeweiligen Punktanzahl, ob wirklich ein Kreis resultiert. Die Generalized Scatter Plots helfen also nur bei der Mustererkennung und bei der Strukturanalyse.

Eine weitere geplante Arbeit liegt in der Anpassung des Exhaustive Pixel Placement - Algorithmus. Bisher werden die sich überdeckenden Datenpunkte kreisförmig um den Ursprungsort herum angeordnet. Dies führt aber zu visuellen Artefakten, welche nicht datengestützt erzeugt werden. Daher soll nun die lokale Datenverteilung in Form der lokalen Korrelation berücksichtigt werden. Diese wird im Bereich der überdeckenden Punkte berechnet und soll sich auf das Pixel Placement auswirken. Somit wird keine kreisförmige Versetzung mehr durchgeführt, sondern ein ellipsoides Pixel Placement. Dabei entspricht die Ellipse bzw. die Stauchung des Kreises der Stärke der Korrelation. Zusätzlich wird die Ellipse in Korrelationsrichtung rotiert, so dass aus der Lage und Form der Ellipse Rückschlüsse auf die lokale Korrelation und damit auch auf die lokale Datenverteilung gezogen werden können.

# Danksagung

Mein allererster Dank gilt meinem Betreuer Herrn Dr. Peter Bak für seine hilfreichen Hinweise, Anmerkungen und nicht zuletzt seine motivierende Gesprächen. Ohne sie wäre diese Masterarbeit in der jetzigen Form nicht denkbar gewesen. Ebenso möchte ich Herrn Geoffrey Ellis für die Erstellung der Sampling-Abbildungen und seinen Hinweisen bezüglich der Evaluierung meinen Dank aussprechen. Zudem möchte ich auch Hewlett Packard und dabei besonders Herrn Umeshwar Dayal und Frau Ming C. Hao für die Möglichkeit danken, als studentische Hilfskraft tätig zu sein. Erst durch die freundliche Genehmigung, beispielsweise den Datensatz des Telefonkonferenzsystems auch für diese Arbeit zu verwenden, war es möglich, die Vorzüge und Nachteile der Genreralized Scatter Plots so realistisch aufzuzeigen. Abschließend möchte ich noch meinen besonderen Dank Herrn Prof. Dr. Daniel A. Keim aussprechen. Nicht nur, dass er mir die Möglichkeit gab, an seinem Lehrstuhl Projekte und Arbeiten durchzuführen, sondern erst durch ihn wurde mir die Möglichkeit eröffnet, für Hewlett Packard zu arbeiten. Die Ergebnisse dieser spannenden und interessanten Tätigkeit sind in dieser Arbeit festgehalten.

# Literaturverzeichnis


[1] P. Bak, M. Schaefer, A. Stoffel, D.A. Keim, and I. Omer. Density Equalizing Distortion of Large Geographic Point Sets. *Cartography and Geographic Information Science*, 36(3):237–250, 2009.

[2] E. Bertini and G. Santucci. Give chance a chance: modeling density to enhance scatter plot quality through random data sampling. *Information Visualization*, 5(2):95–110, 2006.

[3] A. W. Bowman and A. Azzalini. *Applied smoothing techniques for data analysis: the kernel approach with S-Plus illustrations*. Oxford University Press, USA, 1997.

[4] A. W. Bowman and A. Azzalini. Computational aspects of nonparametric smoothing with illustrations from the sm library. *Computational Statistics & Data Analysis*, 42(4):545 – 560, 2003.

[5] J. Bresenham. A linear algorithm for incremental digital display of circular arcs. *Commun. ACM*, 20(2):100–106, 1977.

[6] D. B. Carr, R. J. Littlefield, W. L. Nicholson, and J. S. Littlefield. Scatterplot matrix techniques for large n. *Journal of the American Statistical Association*, 82(398):424–436, 1987.

[7] J. M. Chambers, W. S. Cleveland, B. Kleiner, and P.A. Tukey. *Graphical methods for data analysis*. 1983.

[8] W.S. Cleveland and R. McGill. The many faces of a scatterplot. *Journal of the American Statistical Association*, 79(388):807–822, 1984.

[9] G. Ellis. *Random Sampling as a Clutter Reduction Technique to Facilitate Interactive Visualisation of Large Datasets*. PhD thesis, Lancaster University, 2008.

[10] R Foundation. R – statistical computing tool. http://www.r-project.org/, 2010.

[11] M. Friendly and D. Denis. The early origins and development of the scatterplot. *Journal of the History of the Behavioral Sciences*, 41(2), 2005.

[12] KNIME.com GmbH. knime – konstanz information miner. http://www.knime.org/, 2010.

[13] D.H. House and C.J. Kocmoud. Continuous cartogram construction. In *Proceedings of the conference on Visualization'98*, page 204. IEEE Computer Society Press, 1998.

[14] T.A. Keahey and E.L. Robertson. Techniques for non-linear magnification transformations. In *Proceedings of the IEEE Symposium on Information Visualization, IEEE Visualization*, volume 10, pages 38–45, 1996.

[15] D.A. Keim, M.C. Hao, U. Dayal, H. Janetzko, and P. Bak. Generalized scatter plots. In *Information Visualization Journal (IVS 2009)*. Macmillan Publishers Ltd., 2009.

[16] D.A. Keim, S.C. North, and C. Panse. Cartodraw: A fast algorithm for generating contiguous cartograms. *IEEE Transactions on visualization and computer graphics*, 10(1):95–110, 2004.

[17] D.A. Keim, C. Panse, M. Schafer, M. Sips, and S.C. North. HistoScale: An efficient approach for computing pseudo-cartograms. page 93, 2003.

[18] D.A. Keim, C. Panse, M. Sips, and S. C. North. Pixel based visual data mining of geospatial data. *Computers & Graphics*, 28(3):327 – 344, 2004.

[19] D.A. Keim, C. Panse, M. Sips, and S.C. North. Pixelmaps: A new visual data mining approach for analyzing large spatial data sets. *Data Mining, IEEE International Conference on*, 0:565, 2003.

[20] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, 1967.

[21] U.S. Department of commerce. Census data. http://www.census.gov/, 2010.

[22] U.S. Geological Survey. Latest earthquakes: Feeds & data. http://earthquake.usgs.gov/earthquakes/catalogs/eqs7day-M1.txt, 2010.

[23] A. Unwin, M. Theus, and H. Hofmann. *Graphics of large datasets: visualizing a million (Statistics & computing series)*. Axel Springer Verlag, 2006.