Clutter Measurement and Reduction for Enhanced Information Visualization

by

Natasha Lloyd

A Thesis

Submitted to the Faculty

of the

Worcester Polytechnic Institute

in partial fulfillment of the requirements for the

Degree of Master of Science

in

Computer Science

December 2005

Approved:

Dr. Matthew Ward, Thesis Advisor

Dr. Emmanuel Agu, Thesis Reader

Dr. Michael Gennert, Department Head

Abstract

The effectiveness of information visualization largely depends on the ease and accuracy with which users can access the information. Visual clutter in a display can detract from a user's ability to properly read the information. An ideal visualization needs to maximize the visibility of patterns and structure and minimize the clutter present. Thus far, there has been surprisingly little work done in finding quantitative ways to measure clutter in information visualizations. The goal of this project was to create clutter measurement and reduction techniques that minimize the presence of visual clutter and maximize a user's ability to accurately read the data. These methods were tested and evaluated on a number of visualizations depicting domestic air traffic data.

Contents

1 Introduction **9**

2 Defining Visual Clutter **14**

3 Task Analysis **16**

3.1 Classifying Visualization Tasks 16

3.2 Identifying Air Traffic Visualization Tasks 18

4 Clutter Measurement **21**

4.1 Density 22

4.2 Outliers 28

4.3 Occlusion 31

5 Clutter Reduction **35**

5.1 Information Preserving Methods 35

5.1.1 One-tone Gradient 35

5.1.2 Rainbow Gradient 37

5.1.3 Opacity Gradient 40

5.1.4 Camera Angle Optimization 40

5.2 Information Reducing Methods 41

5.2.1 Altitude Layers 44

5.2.2 Proximity Layers 45

5.3 Remapping Methods 47

5.3.1 Ants Visualization 47

5.3.2 Ants3D Visualization 48

5.3.3 Cityscape Visualization 49

5.3.4 Metaballs Visualization 49

5.3.5 Circles Visualization 50

5.3.6 Sunburst Visualization 50

6 Evaluation Results **52**

7 Conclusions **62**

8 Further Work **63**

References **65**

List of Figures

1 Visualizations used for this project 13

2 Close-up of Ants 3D with bounding boxes. 33

3 One-tone gradient applied to all visualizations 36

4 Rainbow gradient applied to all visualizations 38

5 Opacity gradient applied to all visualizations 39

6 Comparison of features visible with a one-tone gradient versus the features visible with an opacity gradient 41

7 Pre-set views 1-4 used for camera angle optimization 42

8 Pre-set views 5-8 used for camera angle optimization 43

9 Altitude layers filter applied to Ants visualization. 44

10 Altitude layers filter applied to Ants 3D visualization. 45

11 Proximity layers filter applied to Ants visualization. 46

12 Proximity layers filter applied to Ants 3D visualization. 46

13 Proximity layers filter applied to Cityscape visualization. 47

14 Screenshot of the clutter measurement section of the online evaluation survey. 54

15 Amount of clutter present in all visualizations. 55

16 Density measurements. 55

17 Outliers measurements. 56

18 Occlusion measurements. 56

19 Ants: Effectiveness of clutter reduction. 58

20 Ants 3D: Effectiveness of clutter reduction. 58

21 Cityscape: Effectiveness of clutter reduction. 58

22 Metaballs: Effectiveness of clutter reduction. 59

23 Circles: Effectiveness of clutter reduction. 59

24 Sunburst: Effectiveness of clutter reduction. 59

25 Task-Based Evaluation: Locating busy areas. 61

26 Task-Based Evaluation: Identifying traffic routes. 61

27 Task-Based Evaluation: Measuring air traffic volume. 61

1 Introduction

The effectiveness of information visualization largely depends on the ease and accuracy with which users can access the information. Visual clutter in a display can detract from a user's ability to properly read the information. This hindrance can have significant consequences when visualizations are used to make decisions affecting human lives. For instance, visualizations are used to evaluate new air traffic control systems whose aim is to reduce traffic and maintain safe air transportation [1]. These visualizations need to maximize the visibility of patterns and structure and minimize the clutter present.

Visual clutter can take on many forms, and it largely depends on the tasks that are being performed with the visualizations. In this document we will discuss three types of clutter:

- *Density* - the number of objects present relative to the amount of display space available;
- *Outliers* - data points that significantly vary from the majority of all data points; and
- *Occlusion* - objects that either overlap other objects or obstruct other objects from view.

The goal of this project was to create clutter measurement and reduction techniques that minimize the presence of clutter and maximize a user's ability to accurately read the data. The need for better quantitative visualization quality measures has been documented in [13], where the authors posed the following question to the research community:

> "How can we measure the 'goodness' of a particular or combined visualization?"

Although some visualization quality metrics have been proposed in such publications as [21, 22, 23, 29], there has been surprisingly little work done in finding quantitative ways to measure visualization quality. In [3], Brath expanded on the metrics proposed by Tufte in [21, 22, 23] by including issues that affect 3D visualizations. These measurements include data density, the number of dimensions, the number of identifiable points, and redundancy. While these are good as a general framework, they do not always apply as measurements of visualization quality.

Clutter measurement techniques are most useful when designing and evaluating different visualizations. Edward Tufte writes in [22]:

> Clutter and confusion are failures of design, not attributes of information.

This is a widely recognized truth in the field of information visualization and there has been a considerable amount of work done in design techniques that reduce clutter and confusion. Popular clutter reduction methods include similarity clustering [2, 6], shifting from 2D to 3D displays [2, 6], sampling [5], and user interaction [1, 6, 8, 28, 29]. The latter is perhaps the most commonly used method because it gives the user control over which information is more or less visible under the assumption that the user knows what to look for. This is not always a valid assumption, however, since users will find views through trial and error and may miss something that could be important through that process.

Clutter reduction techniques can be grouped into the following three categories:

- Information preserving, which display all data points available and modify some display attributes, such as opacity or camera angle [6], to produce the least cluttered view;
- Information reducing, which remove some data points and aim to find a balance between loss of information and clutter reduction. Such methods of information reduction include filtering/sampling [5, 28], distortion [12, 18], and multi-resolution [8, 26, 27]; and

- Remapping, which maps a data set onto several different visualizations, each of which has its own advantages and disadvantages. Figure 1 shows the sample air traffic data set used in this project mapped to six different visualizations.

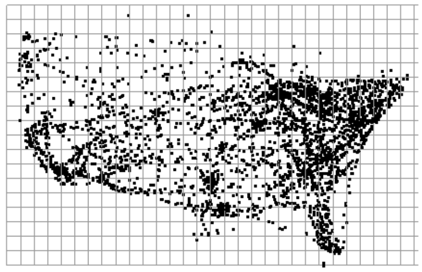
We have developed several different clutter reduction techniques for each of the three categories:

- Information preserving:
  - One-tone gradient
  - Rainbow gradient
  - Opacity gradient
  - Camera angle optimization
- Information reducing:
  - Altitude filtering
  - Proximity filtering
- Remapping:
  - Ants visualization
  - Ants 3D visualization
  - Cityscape visualization
  - Metaballs visualization
  - Circles visualization
  - Sunburst visualization

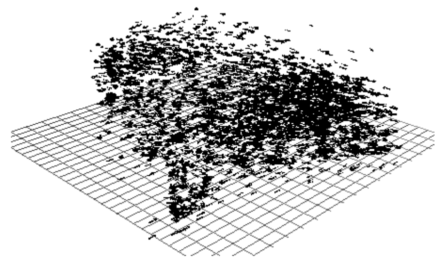
The details of our clutter measurement and reduction techniques are presented in this document. We also present the results of a user evaluation survey that was performed and suggest ideas for further work.

Overview of this Document

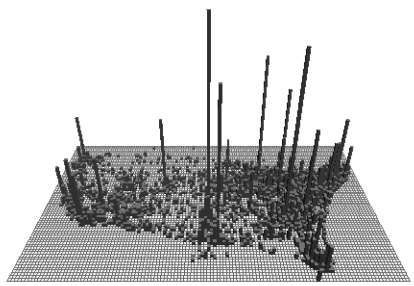
In Chapter 2 we define the term 'clutter' in order to reduce any ambiguities about the meaning of the word. In Chapter 3 we provide an analysis of the sample data set by focusing on the tasks that might be performed with tools visualizing the data. The details of our clutter measurement techniques are presented in Chapter 4. Our clutter reduction methods are described in Chapter 5. Chapter 6 provides an overview of our evaluation techniques as well as some results of that evaluation. In Chapter 7 we present some conclusions based on the work within the scope of this project. Finally, we provide some direction for further work in this topic in Chapter 8.



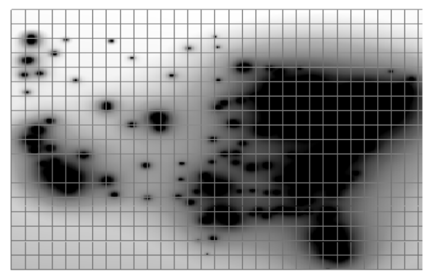
(a) Ants: Every airplane is represented by an icon. Here, the icon is simply a dot.



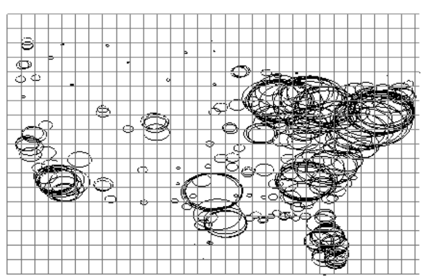
(b) Ants3D: Airplanes are represented as arrows at the latitude, longitude, and altitude location pointing in the direction the airplane is heading.



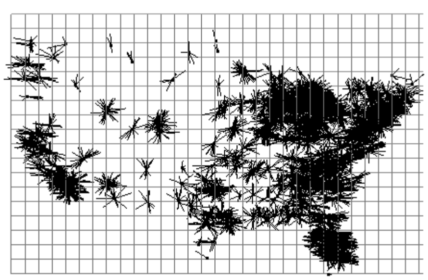
(c) Cityscape: The view is divided into a grid of user-specified size (above, 25 mi^2). The bars represent the volume of traffic at each grid square.



(d) Metaballs: Each active airport is represented by a metaball whose radius represents the volume of traffic at that airport.



(e) Circles: A simplified version of the metaballs visualization, where each active airport is shown as a circle whose radius represents the volume of traffic at that airport.



(f) Sunburst: Every airport is a dot with rays for every moving aircraft within a certain distance of the airport. The rays point in the direction that the aircraft is traveling.

Figure 1: Visualizations used for this project

2 Defining Visual Clutter

Since clutter is a term that is found in many different disciplines, it often holds different meanings. In radar applications, for instance, clutter can be associated with signals or echoes on radar that interfere with desired signals. Things such as mountains, vehicles, water, and birds can cause radar clutter by interfering with signals that are being observed [10]. Another example is verbal clutter, which occurs when there are more words and ideas present than a person can process. This may lead to mental information overload and hinder a person's understanding of the problem at hand [7]. Other examples of clutter exist in fields such as computer vision [9, 14], advertising [17], and Human-Computer Interaction [15].

In information visualization, the definition for clutter remains vague. In some cases, clutter is simply referred to as the number of objects present [27]. However, many visualization researchers also acknowledge that clutter is about more than just density. Edward Tufte [21, 22, 23] discusses clutter as anything that causes confusion in a visual display of information. In fact, according to Tufte, large amounts of data do not always cause clutter; it is, rather, failures in the design of the visual display that can create confusion. Some sources of clutter identified by Tufte include:

- ineffective display of data density
- poor layout decisions
- strong contrast
- bad use of color
- chartjunk, such as extra lines, emphasizing unimportant information, and using unnecessary icons

When we look at how clutter is defined in the above-mentioned examples, and specifically in information visualization, we see that it often depends on personal judgement. What is clutter to one person may not be clutter to another person. There does exists a general idea of clutter, however, which needs to be expressly defined.

Ruth Rosenholtz [15] suggested the following definition of clutter for scientific exploration:

> "Clutter is the state in which excess items, or their representation or organization, lead to a degradation of performance at some task."

This definition is meant for a specific application of the term clutter and relies heavily on the concept of clutter as an excess of something. This may not always be the case in other applications, however. In the radar example mentioned earlier, for instance, clutter has nothing to do with the quantity of objects, but rather with the *types* of objects. In some displays, even outliers that are *sparsely* distributed can create clutter by adding unnecessary information that can confuse the user.

A common thread in all the applications is that clutter is something that causes confusion. It can do so in many ways – by drawing attention to unimportant information, by making it difficult to distinguish individual points, by littering a display with extraneous objects, or by making information difficult to see. The specific ways in which clutter causes confusion vary from one application to another, but the *concept* of clutter remains similar throughout all fields.

Taking this into consideration, I propose the following, more general, definition of clutter, which will be used throughout this document:

> DEFINITION:
> *Clutter is a state of confusion that degrades both the accuracy and ease of interpretation of information displays.*

3 Task Analysis

3.1 Classifying Visualization Tasks

In order to understand which attributes of a visualization can lead to confusion, it is important to understand the types of tasks that users can perform with the visualization. In some cases, visualizations are developed for domain-specific tasks [20]. However, many studies have been done in an effort to evaluate and classify more general types of tasks that users perform with visualization tools.

There have been several different approaches taken to classify tasks. One such approach, proposed by Chuah and Roth [4], looks at tasks performed when accessing and exploring data. Chuah and Roth categorized visualization tasks into three groups:

- *Graphical operations*, which are actions a user can perform on the graphical attributes of a display, such as encoding data through different mappings and transforms and manipulating objects;
- *Set operations*, where the user creates and manipulates sets of objects; and
- *Data operations*, which deal directly with the data being visualized.

Wehrend and Lewis [24] described another possible set of visualization tasks, containing the following steps that a user can perform to analyze a data set:

- *Identify*, where the user describes an item in a data set without previous knowledge;
- *Locate*, where the user finds an item in a data set with previous knowledge of the item;

- *Distinguish*, where a user can see different objects as distinct visual entities;
- *Categorize*, where a user can describe objects as belonging to different categories;
- *Cluster*, where a user can see objects that belong to similar categories grouped together;
- *Distribute*, where a user specifies categories and objects belonging to them are distributed among them;
- *Rank*, where a user is can identify an order to how the objects are displayed;
- *Compare*, where a user compares data items based on their attributes;
- *Associate*, where a user establishes relations between objects displayed; and
- *Correlate*, where a user observes shared attributes among objects.

Other ways to characterize tasks have been explored in works such as [16, 19, 25]. Many of the tasks identified, however, only deal with information retrieval and exploration. Other studies have focused on the thought-process of the users.

In [11] Hibino performed a study with visualization experts, where the subjects were instructed to perform analysis of a data set on tuberculosis using a visualization tool. Although only five people were included in the study, Hibino extracted the following high-level tasks from observing and interviewing the subjects:

- *Prepare*, which includes gathering/learning data background information and other preparation tasks to get the data ready for analysis;
- *Plan*, which includes creating a hypothesis and coming up with a strategy;
- *Explore*, where the user gets familiar with the data set by investigating it in various manners;

- *Present*, which includes the organization or ranking of data;
- *Overlay*, when the user compares various notes and displays to assess his observations;
- *Re-orient*, where goals and progress are reviewed; and
- *Other*, which can include things like gathering statistics.

All the efforts to identify the tasks that users perform when working with a visualization can help with the evaluation of visualizations. If we have a firm grasp on what users want to do, we can then empirically assess the quality and appropriateness of different visualizations.

3.2 Identifying Air Traffic Visualization Tasks

In terms of visual clutter, it is important to be aware of how a visualization will be used when identifying the types of clutter present. Depending on the visualization's purpose, different attributes may cause confusion that degrades the accuracy and ease of interpreting the information presented.

Throughout this document, a sample data set will be presented as a case study for the subject of visualization quality. This data set contains geographical information about air traffic throughout the United States and was provided by the Air Force Research Labs. For each time slice, the data set lists the following information about every aircraft that is currently active:

- A unique flight ID
- Latitude
- Longitude
- Altitude (in meters)

- Heading direction (in degrees)
- Ground speed (in miles per hour)

Defining the types of tasks that may be performed was the first step in the analysis of the sample air traffic data set. Since little guidance was given with regard to how the visualizations will be used, we performed our own evaluation based on three sample visualizations that were provided. We identified the following possible tasks and features of interest in the visualizations:

- Ants:
  - Location of airports/hubs
  - Busy areas without airports
  - Popular destinations and origins
  - Air traffic routes
  - Air traffic volume
  - Overlapping aircraft
  - Effects of time progression
  - Rate of change of traffic volume
- Cityscape:
  - Location of airports/hubs
  - Busy areas without airports
  - Air traffic routes
  - Air traffic volume
  - Effects of time progression
  - Rate of change of traffic volume
- Metaballs:

- Busy areas
- Air traffic volume
- Effects of time progression
- Rate of change of traffic volume

4 Clutter Measurement

Quality measurement metrics are heavily dependent on each specific visualization and on the tasks that are being performed with that visualization. For instance, both Tufte and Brath claim that visualizations with more data presented per square centimeter are more effective. This is true when we assume that the viewer is interested in seeing the whole picture. However, if one's task is to identify certain aspects of individual data points, having clusters of data may prevent this task from being performed effectively.

Based on our definition of clutter, which we developed in section 2, visualization quality is closely tied to the amount of clutter present. Since clutter is defined as a state of confusion that degrades the ease and accuracy of interpretation of information displays, saying that a visualization is cluttered is equivalent to saying that a visualization is of poor quality.

The first step we took in developing clutter measurement techniques for the air traffic data set was to look at several sample visualizations to identify possible tasks and aspects that could prevent the user from performing those tasks. The table below lists the possible sources of clutter we identified and their presence in each of the three sample visualizations.

	Ants	Cityscape	Metaballs
Outliers	X	X	
Density	X	X	
Occlusion	X	X	
Color	X	X	X
Chartjunk	X	X	X

In the table above, the following definitions were used for each source of clutter:

- *Outliers* - data points that significantly vary from the majority of all data points

- *Density* - the number of objects present relative to the amount of display space available

- *Occlusion* - objects that either overlap other objects or obstruct other objects from view

- *Color* - the number of distinct colors present

- *Chartjunk* - elements of a display that are unnecessary and distracting [21]

For this project, we chose to focus on outliers, density, and occlusion, which are discussed in detail below. Both color and chartjunk were taken into consideration when designing new visualizations and improving existing ones, but they were not measured in a quantitative way.

4.1 Density

In general terms, density is the number of data points per unit of space. Depending on the visualization that is being used, the number of objects may vary. We will now go over the density measures used in this project in detail.

The following variables and functions are common to all visualizations:

// updated every frame:
$i \leftarrow$ current frame number
$data \leftarrow$ {all airplanes at frame i}
$grid \leftarrow GET\_FILLED\_GRID(data)$

// global variables, calculated by each measurement technique:
n_{max} // the maximum number of objects present
m // the maximum percentage of the grid that is filled

$GET\_FILLED\_GRID(data)$

```
r ← number of rows in grid
c ← number of columns in grid
for i ← 1 to r
    for j ← 1 to c
        grid[i][j] ← {∅}
for every airplane a ∈ data
    for i ← 1 to r
        for j ← 1 to c
            if location of a is within coordinates of grid[i][j]
            then grid[i][j] ← grid[i][j] ⋃ {a}
return grid
```

$GET\_FILLED\_AIRPORTS(data)$

```
airports ← {all airports}
for every airport p ∈ airports
    p ← {∅}
t ← user defined threshold
for every airport p ∈ airports
    for every airplane a ∈ data
        d ← distance from a to p
        if d < t
        then p ← p ⋃ {a}
return airports
```

Now, we will look at the specific algorithms used to compute the density of each visualization.

Ants and Ants3D:

Here, we count the number of airplanes present and calculate the percentage of screen space that is being used by these airplanes.

$INITIALIZE()$

// performed once when visualization is initialized

$i_{max} \leftarrow$ index when the maximum number of airplanes are present

$data_{max} \leftarrow \{$all airplanes at frame $i_{max}\}$

$n_{max} \leftarrow |data_{max}|$

$grid_{max} \leftarrow GET\_FILLED\_GRID(data_{max})$

$r \leftarrow$ number of rows in $grid_{max}$

$c \leftarrow$ number of columns in $grid_{max}$

$f \leftarrow 0$

for $i \leftarrow 1$ to r

for $j \leftarrow 1$ to c

if $|grid_{max}[i][j]| > 0$

then $f \leftarrow f + 1$

$m \leftarrow \frac{f}{r*c}$

$GET\_DENSITY()$

// normalized to return values ranging from 0 to 10

return $10 * m * \frac{|data|}{n_{max}}$

Cityscape:

For this visualization, we count the number of bars that are displayed and calculate the percentage of the grid that is filled by these bars.

$INITIALIZE()$

// performed once when visualization is initialized

$i_{max} \leftarrow$ index when the maximum number of airplanes are present

$data_{max} \leftarrow$ {all airplanes at frame i_{max}}

$n_{max} \leftarrow 0$

$grid_{max} \leftarrow GET\_FILLED\_GRID(data_{max})$

$r \leftarrow$ number of rows in $grid_{max}$

$c \leftarrow$ number of columns in $grid_{max}$

for $i \leftarrow 1$ to r

for $j \leftarrow 1$ to c

if $|grid_{max}[i][j]| > 0$

then $n_{max} \leftarrow n_{max} + 1$

$m \leftarrow \frac{n_{max}}{r * c}$

$GET\_DENSITY()$

$r \leftarrow$ number of rows in $grid$

$c \leftarrow$ number of columns in $grid$

$n \leftarrow 0$

for $i \leftarrow 1$ to r

for $j \leftarrow 1$ to c

if $|grid[i][j]| > 0$

then $n \leftarrow n + 1$

// normalized to return values ranging from 0 to 10

return $10 * m * \frac{n}{n_{max}}$

Circles and Metaballs:

Both of these visualizations display airports, so the algorithm below counts the number of active airports and approximates the percentage of screen space being used.

$INITIALIZE()$

// performed once when visualization is initialized

$i_{max} \leftarrow$ index when the maximum number of airplanes are present

$data_{max} \leftarrow$ {all airplanes at frame i_{max}}

$n_{max} \leftarrow |\{\text{all airports}\}|$

$grid_{max} \leftarrow GET\_FILLED\_GRID(data_{max})$

$r \leftarrow$ number of rows in $grid_{max}$

$c \leftarrow$ number of columns in $grid_{max}$

$f \leftarrow 0$

for $i \leftarrow 1$ to r

for $j \leftarrow 1$ to c

if $|grid_{max}[i][j]| > 0$

then $f \leftarrow f + 1$

$m \leftarrow \frac{f}{r*c}$

$GET\_DENSITY()$

$airports \leftarrow GET\_FILLED\_AIRPORTS(data)$

$n \leftarrow 0$

for $i \leftarrow 1$ to $|airports|$

if $|airports[i]| > 0$

then $n \leftarrow n + 1$

// normalized to return values ranging from 0 to 10

return $10 * m * \frac{n}{n_{max}}$

Sunburst:

Here we count the number of rays being displayed and approximate the amount of screen space taken up by those rays.

$INITIALIZE()$

// performed once when visualization is initialized

$i_{max} \leftarrow$ index when the maximum number of airplanes are present

$data_{max} \leftarrow$ {all airplanes at frame i_{max}}

$airports \leftarrow GET\_FILLED\_AIRPORTS(data_{max})$

$n_{max} \leftarrow 0$

for $i \leftarrow 1$ to $|airports|$

  $n_{max} \leftarrow n_{max} + |airports[i]|$

$grid_{max} \leftarrow GET\_FILLED\_GRID(data_{max})$

$r \leftarrow$ number of rows in $grid_{max}$

$c \leftarrow$ number of columns in $grid_{max}$

$f \leftarrow 0$

for $i \leftarrow 1$ to r

  for $j \leftarrow 1$ to c

    if $|grid_{max}[i][j]| > 0$

    then $f \leftarrow f + 1$

$m \leftarrow \frac{f}{r*c}$

$GET\_DENSITY()$

$airports \leftarrow GET\_FILLED\_AIRPORTS(data)$

$n \leftarrow 0$

for $i \leftarrow 1$ to $|airports|$

  $n \leftarrow n + |airports[i]|$

// normalized to return values ranging from 0 to 10

return $10 * m * \frac{n}{n_{max}}$

From the pseudocode above, we can see that the way density is computed is similar in all visualizations. The differences come in counting the number of objects present because each visualization displays different types of objects. For instance, Ants displays individual airplanes, while Cityscape displays bars representing air traffic over a certain area. In general, however, the density always represents the amount of display space covered by the objects displayed. Although this is not exactly the definition of density that is normally used, it does give a good approximation that mostly agrees with user opinions (see Section 6).

4.2 Outliers

Outliers are defined as data points that significantly vary from the majority of all data. Similar to density, the way outliers are measured is dependent on the type of visualization that is being used. In all cases, outlier calculation requires some user-defined threshold that represents the deviation point when a data point becomes an outlier. For instance, we may want outliers to be airplanes that are at least distance x away from other airplanes. We will now discuss the outliers measurements that were used for this project by looking at individual visualizations.

(See section 4.1 for variables and functions that are common to all visualizations.)

Ants and Ants3D:

Several possibilities for measuring outliers were considered for these two visualizations. The first possibility was a nearest-neighbor approach, where an object would be considered an outlier if its nearest neighbor was too far away (as defined by the user). While this approach would certainly find some outliers, it would not take into consideration cases where two airplanes are traveling close to each other but far away from all other airplanes. To a user, these airplanes would most likely appear as outliers, but the measurement method would not classify them as such.

The second possibility was a density-based approach, where an object would be considered an outlier if it had too few neighbors (again, as defined by the user). This approach takes care of the problem with the nearest-neighbor method, but could be computationally expensive since it would have to calculate the number of neighbors for each object. This would take O(n^2) time because each object would need to be compared with every other object.

The third possibility, and the one that was selected for this project, is an optimization on the density-based approach, which utilizes the grid that is already used in the visualizations. In this approach, the screen is divided into a grid of user-defined size and the grid is filled in with airplanes that belong to each grid square. Then, the algorithm looks at each grid square to see how many airplanes are present and if the number is less than a user-defined threshold, all airplanes in that square are considered outliers. This approach is a bit faster than the pure density-based approach because the size of the grid is generally smaller than the number of airplanes and so fewer comparisons need to be made. Although this method has the added cost of filling the grid, this is negligible since the grid is filled at every frame anyway for other purposes, such as determining the color of airplanes. It should be noted, however, that it is possible for this approach to generate somewhat inaccurate results if adjacent grid locations add up to a cluster even though individually they may have been outliers.

GET_NUM_OUTLIERS()

$r \leftarrow$ number of rows in *grid*

$c \leftarrow$ number of columns in *grid*

$n \leftarrow 0$

$t \leftarrow$ user-defined threshold (number of airplanes per grid square)

for $i \leftarrow 1$ to r

for $j \leftarrow 1$ to c

if $0 < |grid[i][j]| < t$

then $n \leftarrow n + |grid[i][j]|$

// normalized to return values ranging from 0 to 10

return $10 * \frac{n}{|data|}$

Cityscape:

For this visualization, an outlier is a grid square that has too few airplanes over it. The number returned by the algorithm before is the fraction of outliers to the total grid space available.

GET_NUM_OUTLIERS()

$r \leftarrow$ number of rows in *grid*

$c \leftarrow$ number of columns in *grid*

$n \leftarrow 0$

$t \leftarrow$ user-defined threshold (number of airplanes per grid square)

for $i \leftarrow 1$ to r

for $j \leftarrow 1$ to c

if $|grid[i][j]| < t$

then $n \leftarrow n + 1$

// normalized to return values ranging from 0 to 10

return $10 * \frac{n}{r*c}$

Circles, Metaballs, and Sunburst:

All three of these visualizations display airports, so here, outliers are airports with too few airplanes within a user-specified distance of them. This distance is specified in the function

$GET\_FILLED\_AIRPORTS()$ (see section 4.1).

$GET\_NUM\_OUTLIERS()$

$n \leftarrow 0$

$a \leftarrow 0$

$airports \leftarrow GET\_FILLED\_AIRPORTS(data)$

$t \leftarrow$ user-defined threshold (number of airplanes per airport)

for $i \leftarrow 1$ to $|airports|$

if $|airports[i]| > 0$

then $a \leftarrow a + 1$

if $|airports[i]| < t$

then $n \leftarrow n + 1$

// normalized to return values ranging from 0 to 10

return $10 * \frac{n}{a}$

The definitions of outliers used above are just one set of possibilities. Depending on the tasks that a user is trying to perform with a visualization, it may make sense to redefine outliers to get a different measurement. For instance, in the Circles, Metaballs, and Sunburst visualizations, outliers could be defined as airports that are not close to any other airports. In other words, an outlier is a term that is relative to the aspect of the data that is most important to the user.

4.3 Occlusion

Occlusion occurs when objects obstruct other objects from view in both 2D and 3D displays. In 2D occlusion is caused by overlapping objects, while in 3D it is caused by objects that are closer to the camera covering objects that are further away. In both cases, accurate occlusion measures are costly and not practical for animated visualizations. Because of this we created occlusion measures that are approximations of the total occlusion present. For the purposes of most clutter reduction, the estimates calculated by our technique are good enough to judge the quality of a visualization.

The pseudocode below describes how occlusion measurement was performed for the different visualizations.

2D Visualizations:

For Ants, Circles, and Sunburst, occlusion was measured based on the distance between objects. If two objects are too close, then one of them is occluded. The specific objects vary between the visualizations, but in all cases, a 2D point location can be used to identify the objects, so we will only present one version of the pseudocode to give a general idea. For Ants, the point used is the latitude/longitude location of each aircraft; for Circles, it is the center of each circle; and for Sunburst, it is the location of each airport. Although Metaballs is a 2D visualization, the occlusion measurements do not apply to it because there is no occlusion possible in this visualization.

$GET\_NUM\_OCCLUSIONS()$

$n \leftarrow 0$

for every object a_1

for every other object a_2

if $DISTANCE(a_1, a_2) > \frac{SIZEOF(a_1)}{2} + \frac{SIZEOF(a_2)}{2}$

$n \leftarrow n + 1$

return n

3D Visualizations:

Occlusion measurement for 3D visualizations is a bit more complex than its 2D counterpart. Every object in a 3D visualization can be represented with a bounding box. For Cityscape, the bars themselves are bounding boxes. For Ants 3D, each airplane can be defined with a box that is centered at the location of the airplane and completely encloses it (see Figure 2).

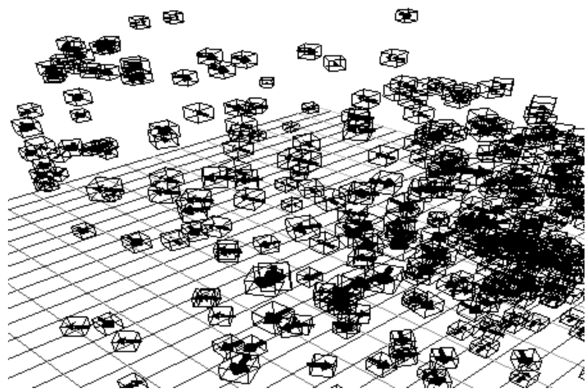


Figure 2: Close-up of Ants 3D with bounding boxes.

Using these bounding boxes, our occlusion measurement extends a ray from the camera to the XZ-plane through the center of each bounding box and checks for other boxes that are close to or intersect that ray. It does this in a way similar to the 2D occlusion measure by checking the distance from each box to the ray, but instead of checking the distance between two points, this method checks the distance between a point and a line. The pseudocode for this method follows.

$GET\_NUM\_OCCLUSIONS()$

$c \leftarrow$ camera position

$n \leftarrow 0$

for every object a_1

// calculate the point of intersection with XZ-plane

$v \leftarrow a_1 - c$

$t \leftarrow \frac{a_1.y}{v.y}$

$p \leftarrow a_1 + t * v$

$l \leftarrow$ line from a_1 to p

for every other object a_2

if $DISTANCE(l, a_2) < \frac{SIZEOF(a_2)}{2}$

then $n \leftarrow n + 1$

return n

As mentioned earlier, this is just an approximation of the actual number of occlusions.

5 Clutter Reduction

Since the quality of a visualization is based on how easily a user can obtain useful information from the display, it is important to develop techniques that reduce the clutter present. Generally, these techniques fall into three categories:

- *Information Preserving* - all objects are displayed on the screen and attributes such as color, opacity, and camera angle are used to reduce the amount of clutter present.
- *Information Reducing* - some objects may not be displayed and data may be altered in order to reduce clutter.
- *Remapping* - data is visualized in several different ways, with each mapping having its own advantages and disadvantages.

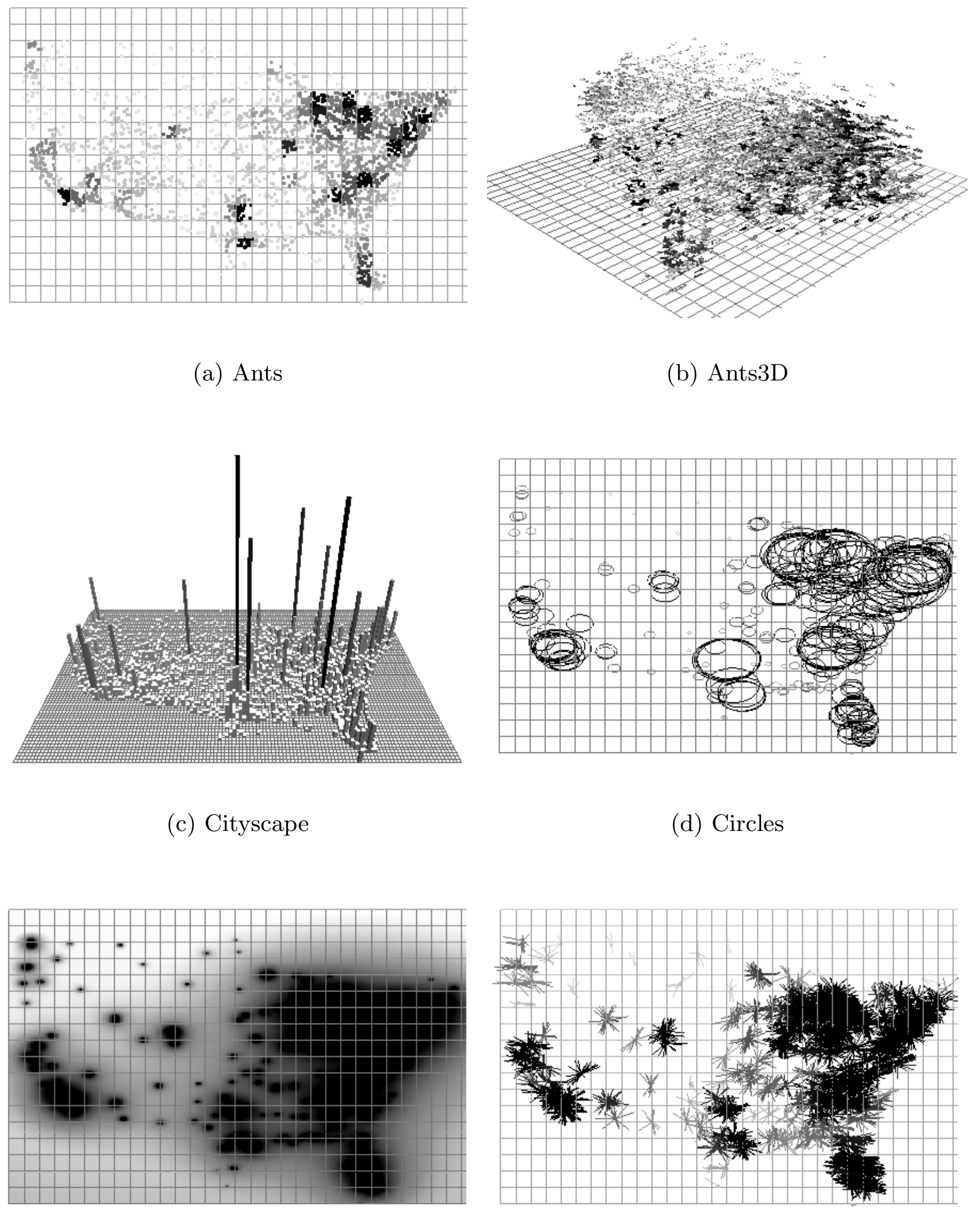
We have developed clutter reduction methods that fall into all three categories and we discuss them in detail below.

5.1 Information Preserving Methods

Methods that preserve information do not change any attributes of the original data. For this project, we developed four such methods, which include using color gradients, an opacity gradient, and a camera angle optimization technique.

5.1.1 One-tone Gradient

A one-tone gradient (see Figure 3) is a range of colors that blend the foreground color into the background color. In our visualizations, the one-tone gradient represented the following:



(a) Ants
(b) Ants3D
(c) Cityscape
(d) Circles
(e) Metaballs
(f) Sunburst

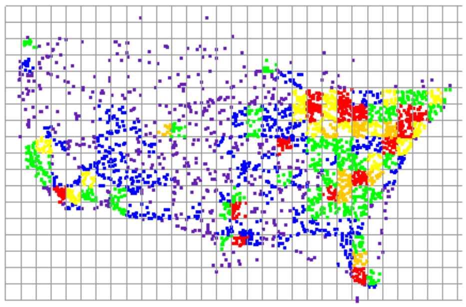
Figure 3: One-tone gradient applied to all visualizations

- *Ants and Ants3D* - Airplanes in heavy traffic grid squares are colored brighter (closer to the foreground color) than those in lighter traffic grid squares. The grid system was used to speed up computation.
- *Cityscape* - Bars that represent grid squares with heavy traffic were colored brighter than those with lighter traffic.
- *Circles, Metaballs, and Sunburst* - Airports that have more airplanes within some user-defined distance of them are colored brighter than those that have fewer airplanes in their proximity.

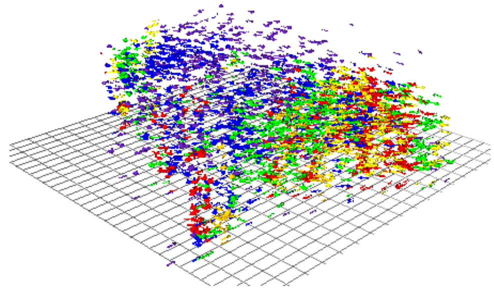
5.1.2 Rainbow Gradient

A rainbow gradient (see Figure 4) is similar to the one-tone gradient because it assigns color to objects based on some attribute. However, instead of blending from the foreground color to the background color, this technique applies a rainbow color scheme using these colors (in order): red, orange, yellow, green, blue, purple. In our visualizations, the rainbow gradient represented the following:

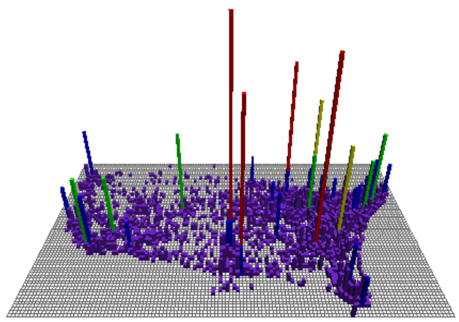
- *Ants and Ants3D* - Airplanes are colored based on the traffic volume in the square where they belong (red is the highest traffic, purple is the lowest).
- *Cityscape* - Bars are colored based on the traffic volume in the square they represent (red is the highest traffic, purple is the lowest).
- *Circles, Metaballs, and Sunburst* - Airports are colored based on the traffic volume in their proximity (red is the highest traffic, purple is the lowest).



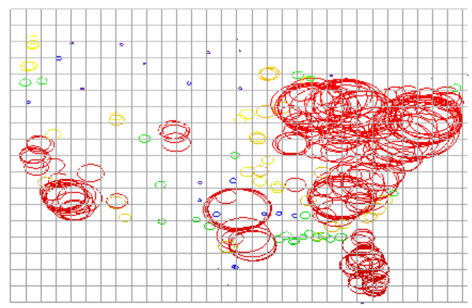
(a) Ants



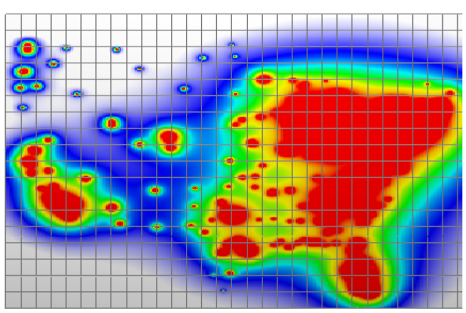
(b) Ants3D



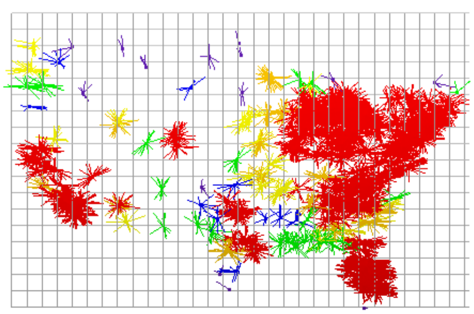
(c) Cityscape



(d) Circles

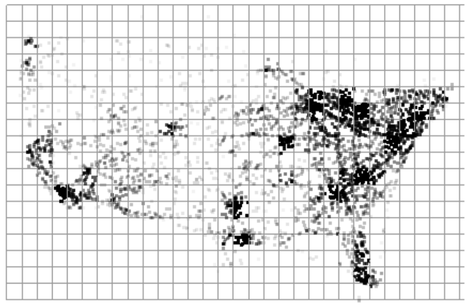


(e) Metaballs

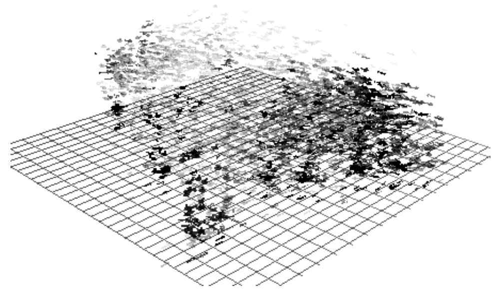


(f) Sunburst

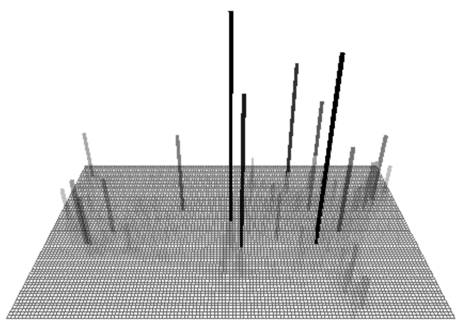
Figure 4: Rainbow gradient applied to all visualizations



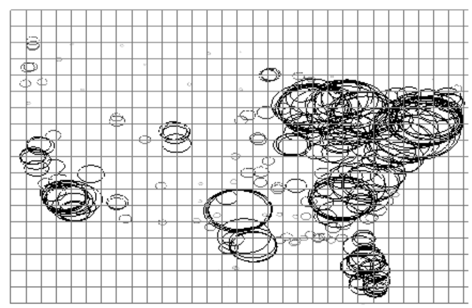
(a) Ants



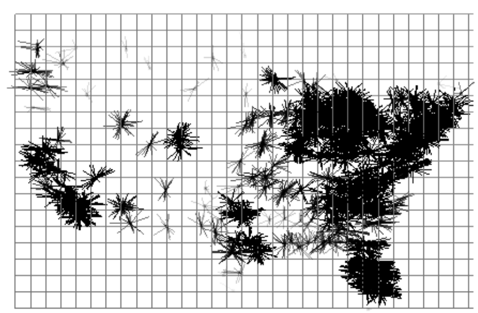
(b) Ants3D



(c) Cityscape



(d) Circles



(e) Sunburst

Figure 5: Opacity gradient applied to all visualizations

5.1.3 Opacity Gradient

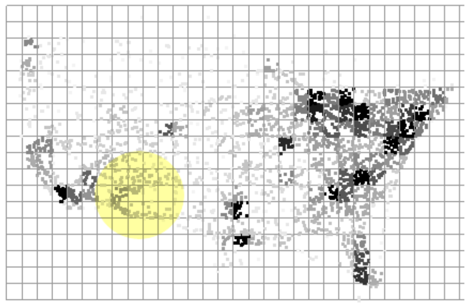
Unlike the one-tone and rainbow gradients, an opacity gradient (see Figure 5) varies the opacity of each object rather than the object's color. This can be used in addition to the color gradients or on its own. In our visualizations, the opacity gradient represented the following (this gradient does not apply to the Metaballs visualization):

- *Ants and Ants3D* - Airplanes in heavy traffic grid squares are more opaque than those in lighter traffic grid squares.
- *Cityscape* - Bars that represent grid squares with heavy traffic are more opaque than those with lighter traffic.
- *Circles and Sunburst* - Airports that have more airplanes within some user-defined distance of them are more opaque than those that have fewer airplanes in their proximity.

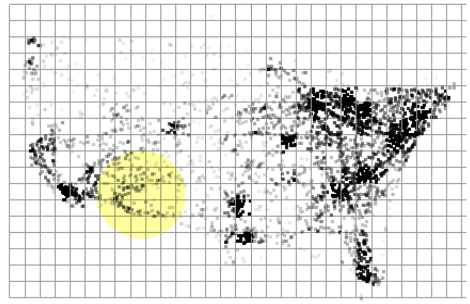
Although the opacity gradient produces a similar effect to the one-tone gradient, it does make certain features more visible. For instance, in the highlighted sections of Figure 6, you can clearly see more patterns in the image with an opacity gradient applied than the image with the one-tone gradient.

5.1.4 Camera Angle Optimization

In order to reduce occlusion in 3D visualizations, we explored a camera angle optimization algorithm. This algorithm uses the occlusion measure discussed in section 4.3 to approximate the amount of occlusion in a small set of pre-defined camera angles and selects the angle with the least occlusion. We chose eight camera angles for this project, but this can easily be expanded.



(a) One-tone Gradient



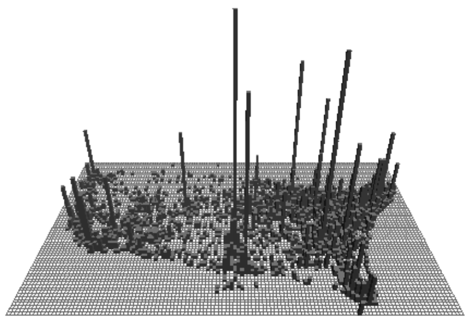
(b) Opacity Gradient

Figure 6: Comparison of features visible with a one-tone gradient versus the features visible with an opacity gradient

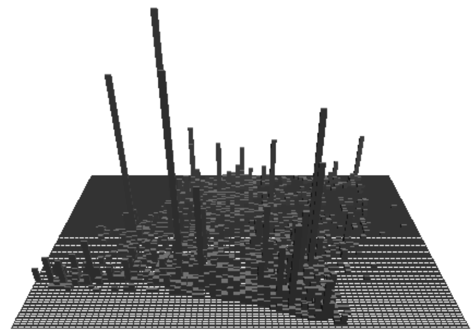
It is important to limit the angles that can be selected in order to avoid choices that may have the least occlusion, but also show the least information. For instance, in the Cityscape visualization a top-down view may have the least occlusion, but it also loses all the height information which is the essence of this visualization. Figures 7 and 8 show the eight camera angles that were available for selection by the algorithm.

5.2 Information Reducing Methods

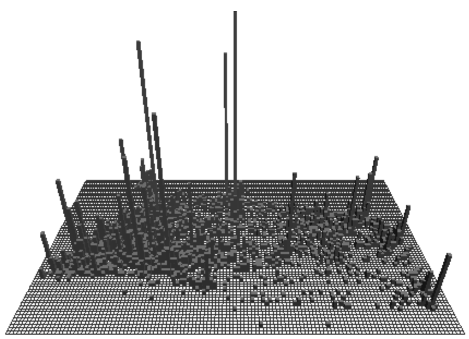
Another option for reducing clutter is to reduce the number of objects that are present in the display with techniques such as clustering, sampling, and filtering. Clustering groups objects that are close together and represents them in a way that helps the user identify that more than one data object is present. Sampling techniques attempt to find data points that represent the majority of the information within the entire data set. Filtering enables users to specify the subset of data in which they are most interested and displays only that subset. For all of these techniques, it is important that the user has control over and is aware of the type of information reduction that is being performed. These methods try to achieve a good balance between the amount of clutter present and the amount of information displayed.



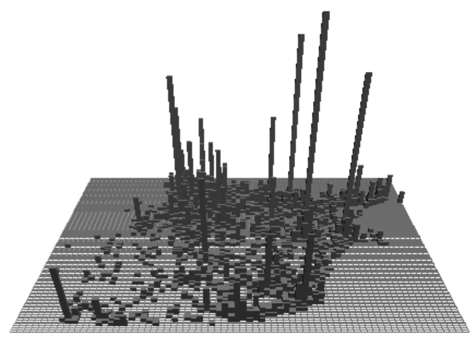
(a) View 1



(b) View 2

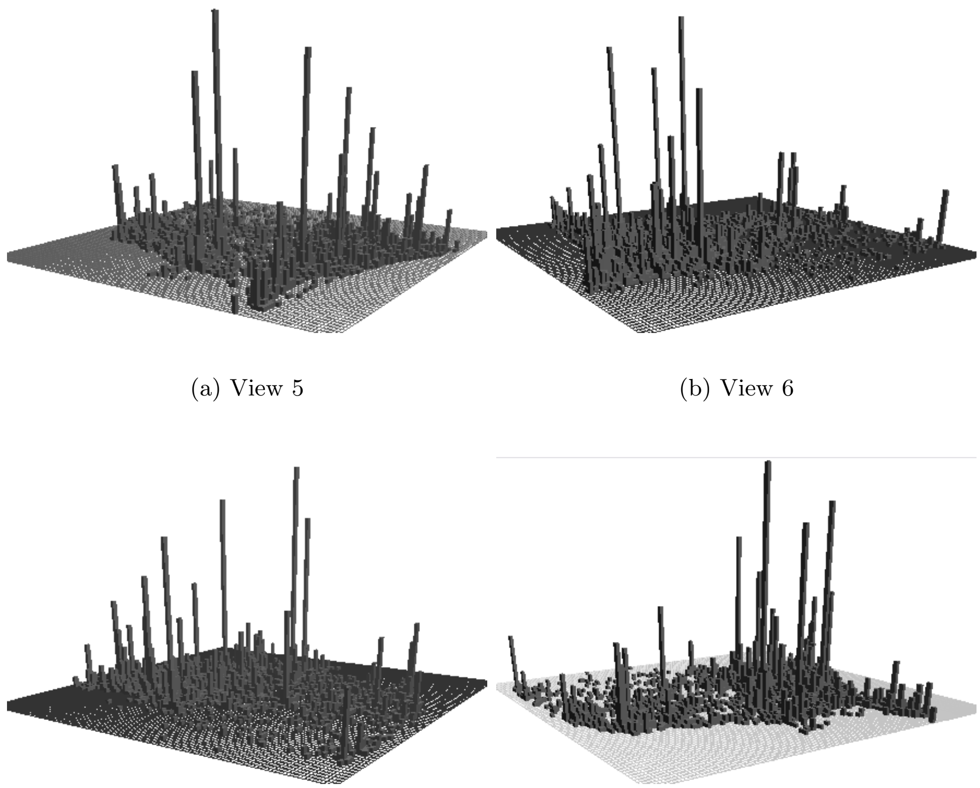


(c) View 3



(d) View 4

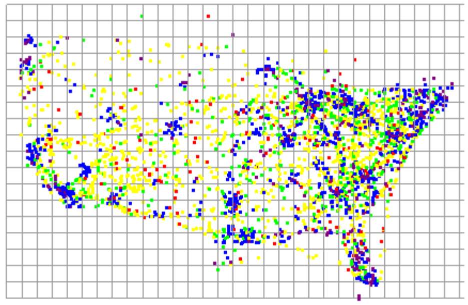
Figure 7: Pre-set views 1-4 used for camera angle optimization



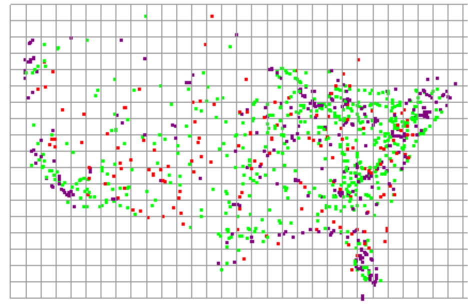
(a) View 5 (b) View 6

(c) View 7 (d) View 8

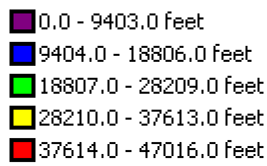
Figure 8: Pre-set views 5-8 used for camera angle optimization



(a) All layers



(b) Selected layers



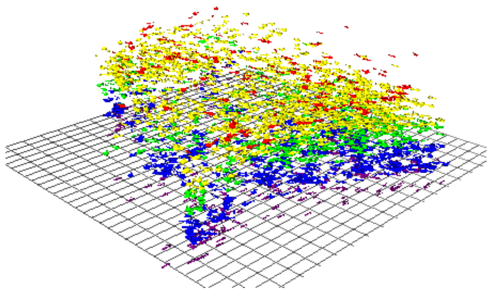
(c) Key

Figure 9: Altitude layers filter applied to Ants visualization.

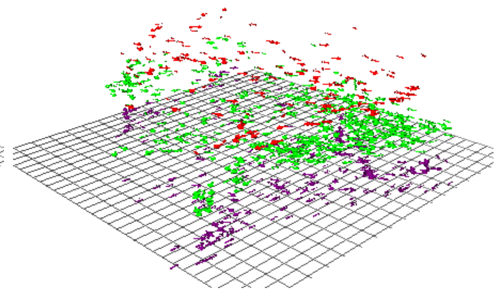
Due to time constraints, we only focused on filtering techniques for this project. We developed two types of filters, which were implemented as layers. If a filtering technique is selected, all objects are divided into layers based on a certain attribute, such as altitude. The user then has control over which layers to display. Each filtering technique is described in detail below.

5.2.1 Altitude Layers

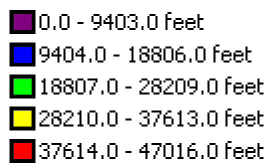
If the altitude layers option is selected, airplanes are grouped into 5 groups based on their altitude. Each group, or layer, has a color associated with it and the user can select which altitudes to display. This technique was only implemented for the Ants (Figure 9) and Ants 3D (Figure 10) visualizations because they display individual aircraft. A possible extension on this could be done with the Cityscape visualization, where each bar could be a stack of



(a) All layers



(b) Selected layers



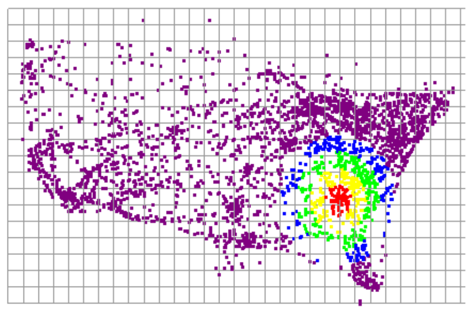
(c) Key

Figure 10: Altitude layers filter applied to Ants 3D visualization.

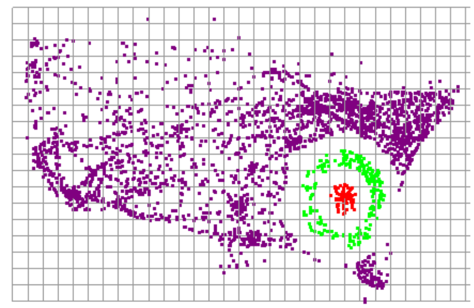
smaller bars representing the number of airplanes at each altitude layer.

5.2.2 Proximity Layers

Proximity layers are similar to the altitude layers, but objects are grouped based on their proximity to some point. This is useful for coloring objects that are close to a certain airport. This technique was applied to the Ants (Figure 11), Ants 3D (Figure 12), and Cityscape (Figure 13) visualizations, which show objects other than airports.



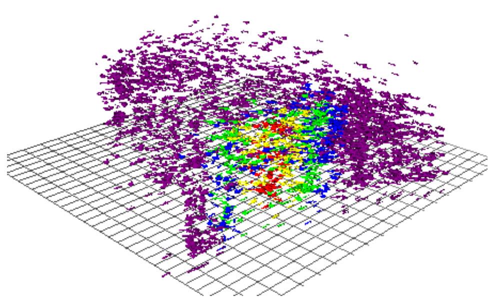
(a) All layers



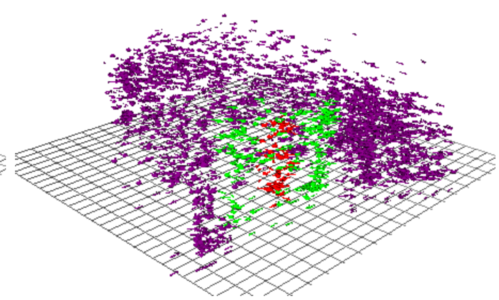
(b) Selected layers

(c) Key

Figure 11: Proximity layers filter applied to Ants visualization.



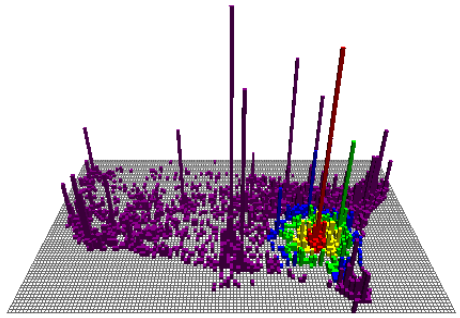
(a) All layers



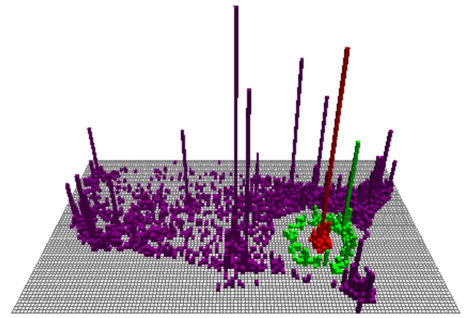
(b) Selected layers

(c) Key

Figure 12: Proximity layers filter applied to Ants 3D visualization.



(a) All layers



(b) Selected layers

(c) Key

Figure 13: Proximity layers filter applied to Cityscape visualization.

5.3 Remapping Methods

The same data set can be visualized in several different ways. Each visual mapping results in a different level and distribution of clutter. In the sections below, we discuss each mapping in more detail. See Figure 1 for images of all the visualizations.

5.3.1 Ants Visualization

In an Ants visualization, each airplane is represented as an icon on a 2D plane that is divided into a grid. For this project, each airplane was a dot, but this can be extended to display more descriptive icons, such as small images of the types of airplanes.

This visualization is very intuitive because it shows the geo-spatial information of air traffic in a way that people are used to seeing. The display is essentially a projection of airplanes onto a map of the country. This makes the visualization easy to read and easy to understand.

However, since all airplanes are displayed, the potential amount of clutter is high. At the busiest time in the sample dataset, there are 3934 airplanes flying over the United States. This is a lot of visual information and some details about individual airplanes may be lost.

5.3.2 Ants3D Visualization

In many ways, Ants 3D is similar to Ants. All airplanes are displayed as icons over a grid of the country. However, in addition to the latitude/longitude coordinates, the altitude of airplanes is also displayed, creating a 3D view of air traffic. In our visualization, we chose to represent airplanes as arrows pointing in the direction the airplane is traveling.

This visualization presents the most realistic view of air traffic because it shows the actual locations of airplanes. Unfortunately, the 3D aspect makes the view very cluttered since users cannot easily distinguish objects that are close by from objects that are far away. When many airplanes are present, it is nearly impossible to get any information from this visualization.

It is not completely useless, however. There is great potential for this visualization if it is used on a smaller scale with sparse data. For instance, this would be useful if we focus on just one airport and track only the airplanes entering and leaving the airport. In this case, the 3D information would be very useful for air traffic control and for evaluation of air traffic control methods around airports.

5.3.3 Cityscape Visualization

The Cityscape visualization is a 3D histogram of air traffic. It divides the country into a grid of user-specified size and shows a bar for each grid square that has airplanes flying over it. This visualization shows the air traffic distribution throughout the country.

With Cityscape, the 3D problems of Ants 3D are solved by having the bars clearly extend from grid squares. This makes it easier to see spatial information and creates a less cluttered view. Also, since each bar represents a section of the grid and not individual airplanes, there are fewer objects present. However, occlusion is an issue in this visualization because of its 3D nature. Bars that are closer to the camera often hide shorter bars, which reduces the accuracy of a user reading the information. This visualization is useful for tasks that do not require information about individual aircraft.

5.3.4 Metaballs Visualization

The Metaballs visualization represents air traffic at each airport as a metaball whose size depends on the volume of traffic around the airport. Airports with more traffic within their radius are represented by larger metaballs than those with little traffic. We chose to have a metaball for each airport in order to limit the number of metaballs we need to render. However, this can easily be changed to have a metaball for each grid square or each aircraft.

This visualization provides a good overall picture of air traffic, but it is difficult to see any details. Although it is useful for identifying patterns in air traffic volume, it takes a very long time to render and thus makes it difficult to animate over time.

5.3.5 Circles Visualization

The Circles visualization is similar to Metaballs. However, instead of using metaballs to represent airports, this visualization uses circles whose radius depends on the amount of air traffic near the airport. This has several advantages over Metaballs. For instance, it is possible to see individual airports, since the circles are not filled in. Also, this visualization is quicker to render than metaballs, though it does slow down as air traffic increases. However, unlike Metaballs, this visualization has higher potential for clutter. Since lines are used to outline the circles, clutter can be created when many circles overlap in one area.

5.3.6 Sunburst Visualization

The Sunburst visualization is a bit more complex than the previously mentioned visualizations. In this visualization, we start by drawing dots at every airport that has airplanes within some user-defined distance of it. Then, for each airport, we draw rays for every airplane in the proximity of the airport. The ray points in the direction that the airplane is heading. It is important to note that the ray *does not* point from the airport to the airplane. The main goal of this visualization is to make routes and popular destinations more apparent. Because of this, it does not make sence to simply draw a line from an airplane to the airport. Instead, we show directional information. The length of the ray is determined by how far the airplane is from the airport.

This visualization can potentially get very cluttered because of number of objects being drawn. Since multiple airports can be very close to each other, it is possible for airplanes to be counted and drawn as rays multiple times, once for each airport. However, since the rays would point in the same direction, this is not really a problem. Rays from airports that are close to each other would just overlap in many cases.

The main benefit of this visualization is that it makes it easier to see flight patterns. Although it currently only shows one starburst icon per airport, it can be extended to show an icon at every grid square to show more information.

6 Evaluation Results

Computer Science students and faculty were asked to participate in an online survey that contained questions about visual clutter in the six visualizations used for this project. With this survey we collected user opinion of clutter measurement, the effectiveness of clutter reduction, and the ability to perform tasks using different visualizations.

The survey started by giving an introduction to the project and the problem statement. It then collected some demographic information, such as the user's level of education and level of expertise in areas such as computer graphics, simulation, and air traffic control. Next, users were provided with explanations of all the visualizations that are used throughout the survey. There were two main sections of the survey, clutter measurement/reduction and task-based evaluation.

There were a total of 27 participants who completed the survey. The demographics information collected is presented in the tables below.

Level of Education

Some College:	13
Bachelor's Degree:	4
Master's Degree:	8
PhD:	2

Subject Familiarity

	Graphics	Visualization	Air Traffic Control	Simulation
None	0	0	8	4
Beginner	6	18	19	15
Intermediate	17	5	0	8
Expert	4	4	0	0

Color Perception Deficiency: 3

Clutter Measurement:

The purpose of this section was two-fold. First, we wanted to compare users' perception of clutter to our measurement techniques. Second, we wanted to assess the effectiveness of our clutter reduction methods. To achieve this, the users were given a definition of clutter and were asked to rate images of visualizations based on how cluttered they were in terms of density, outliers, and occlusion from 1 (least cluttered) to 10 (most cluttered). Figure 14 shows a screen shot of what the users were asked to do.

Users were shown three images of each visualization and each clutter reduction technique that had different volumes of traffic present. This was done to put the images into context of a changing dataset. Since the images were still, showing light, medium, and heavy traffic volumes provided more information about how the images would actually be seen and used.

Figure 15 shows the overall clutter measured by users and by the methods described in Section 4. User measurements mostly agreed with our algorithms' measurements, with a few exceptions. For instance, the Metaballs visualization was perceived as much more cluttered than the measured values. In our algorithms, the Metaballs visualization does not have any occlusion and the density and outliers measures are based on the number of airports. To users, however, this visualization may seem much more cluttered because it covers most of the display space and seems to contain a lot of information.

Figures 16, 17, and 18 show the measurements in more detail and include the trend lines for easier comparison. From these figures, we can see that the density measurement developed for this project is very close to user's perception of density as clutter. The major disparity is in the Metaballs visualization. This is likely a result of users' perception of the mostly filled image as clutter and the actual measure only counting individual airports without consideration of the air traffic volume at the airports. This can be fixed by integrating

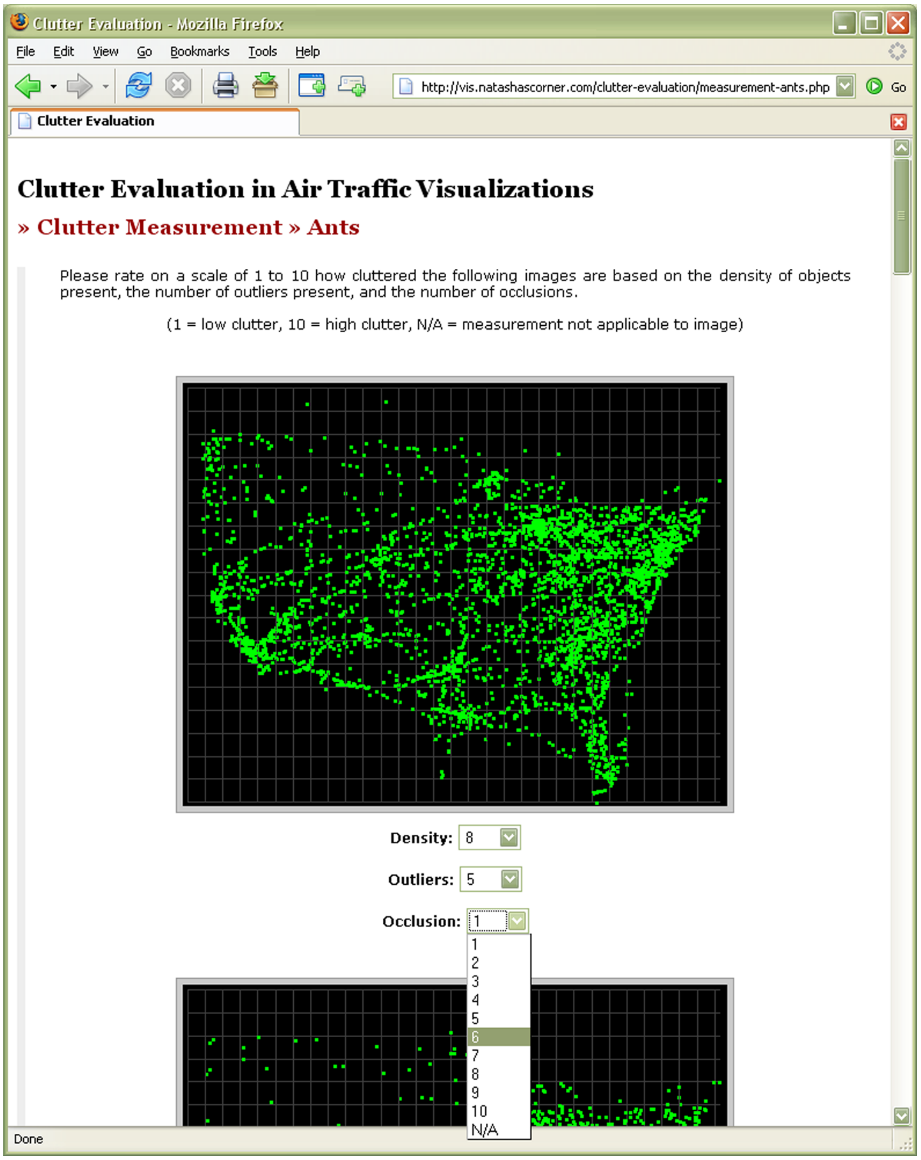


Figure 14: Screenshot of the clutter measurement section of the online evaluation survey.

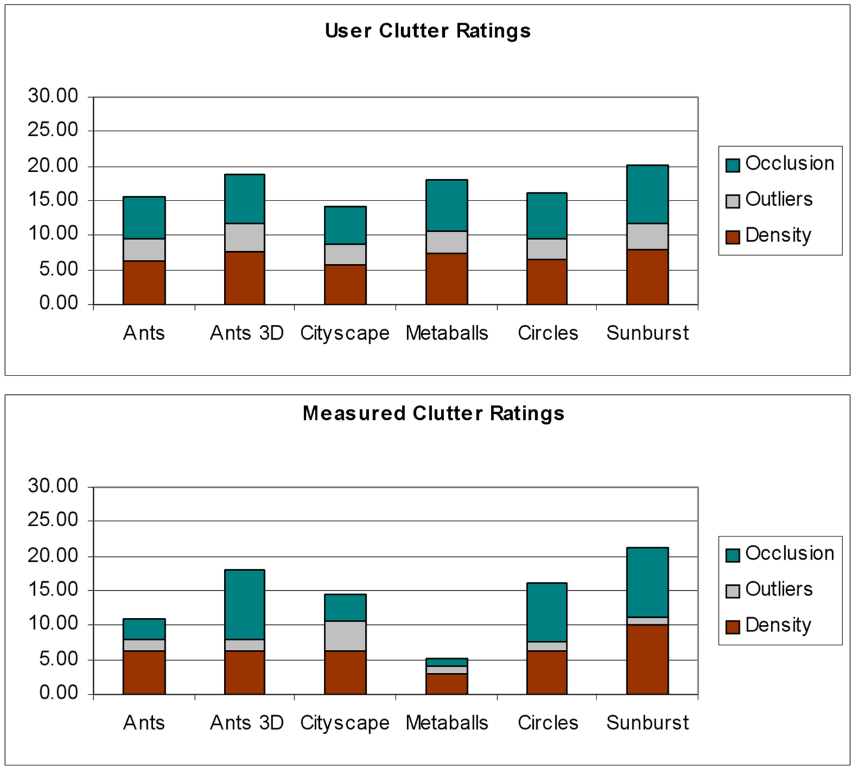


Figure 15: Amount of clutter present in all visualizations.

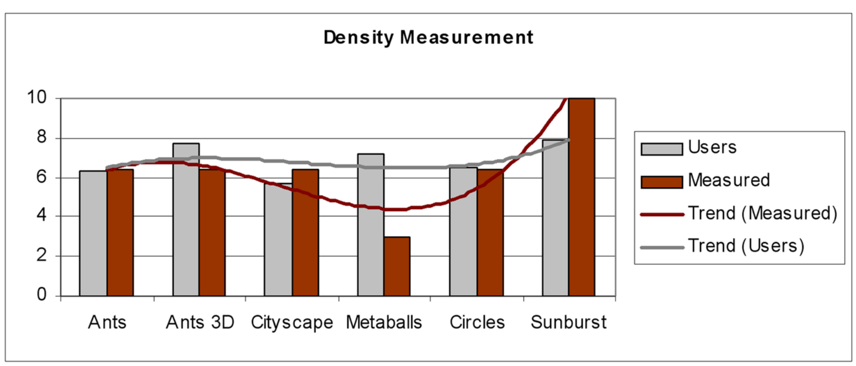


Figure 16: Density measurements.

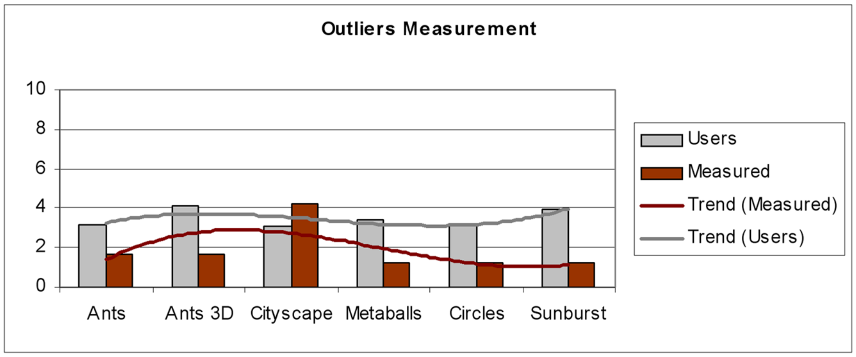


Figure 17: Outliers measurements.

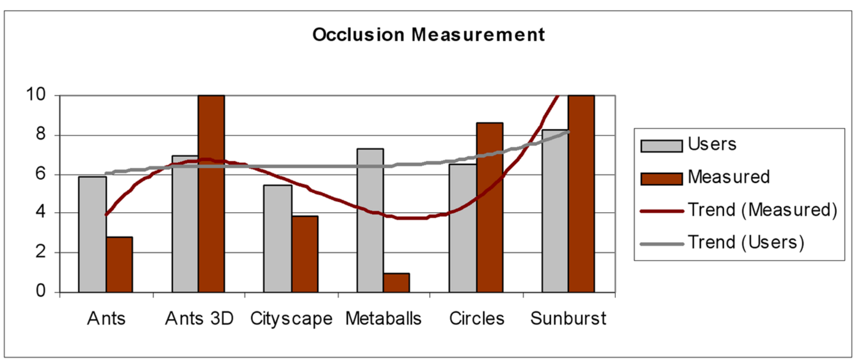


Figure 18: Occlusion measurements.

traffic volume into the density measurements for the Metaballs visualization.

There was more disparity in the outliers measurements, with the calculated clutter being generally lower than the perceived clutter. However, as the trend lines in Figure 17 indicate, the calculations are overall very similar to users' perception. This can easily be calibrated by using a different threshold value to determine outliers.

Occlusion measurements are interesting because there seem to be significant differences between perceived and calculated values. This is likely a result of the nature of occlusion. By definition, occlusion occurs when one cannot see something because it is blocked by other objects. Therefore, the users were actually being asked to guess how much information was hidden from them. If a user sees one dot on the screen, they can approximate how many other dots may be hidden in that spot based on the surrounding density, but there is no way to distinguish how many dots are actually hidden. Our occlusion calculation algorithms, on the other hand, could obtain accurate measurements because they have access to all the data, even data that is hidden from the user. In this case, it is best to rely on the calculated numbers without adjustments since user measurements cannot be considered accurate.

Clutter Reduction:

Figures 19-24 show how users rated the clutter for each visualization with various clutter reduction techniques applied. It is interesting to note that for the Metaballs, Circles, and Sunburst visualizations, the users perceived little to no difference when the various clutter reduction techniques were used. In the other visualizations, we can see the expected result that in most cases images with clutter reduction applied were perceived less cluttered than those without any clutter reduction. It is only in the Cityscape visualization where the single-color gradient was thought to actually add clutter to the visualization.

Task-Based Evaluation:

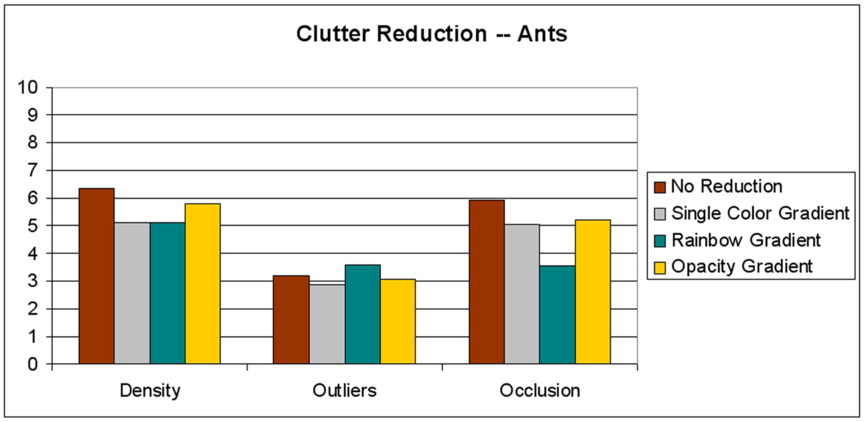


Figure 19: Ants: Effectiveness of clutter reduction.

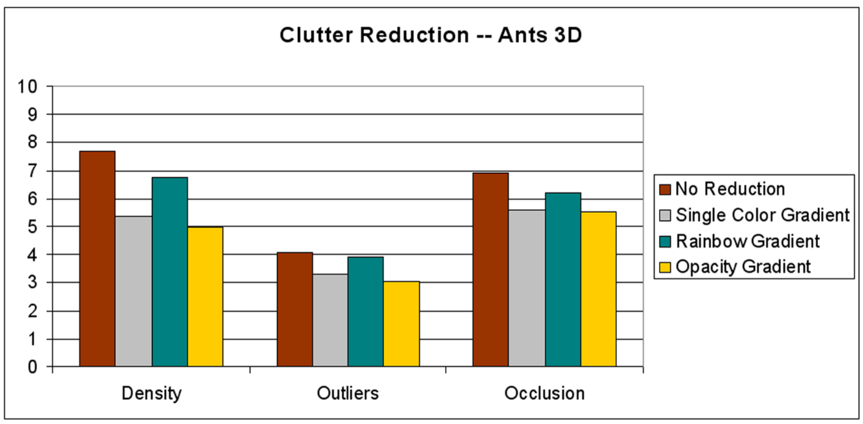


Figure 20: Ants 3D: Effectiveness of clutter reduction.

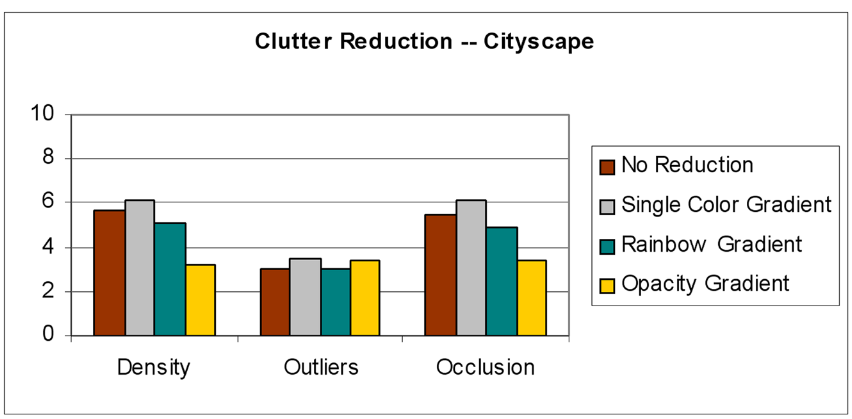


Figure 21: Cityscape: Effectiveness of clutter reduction.

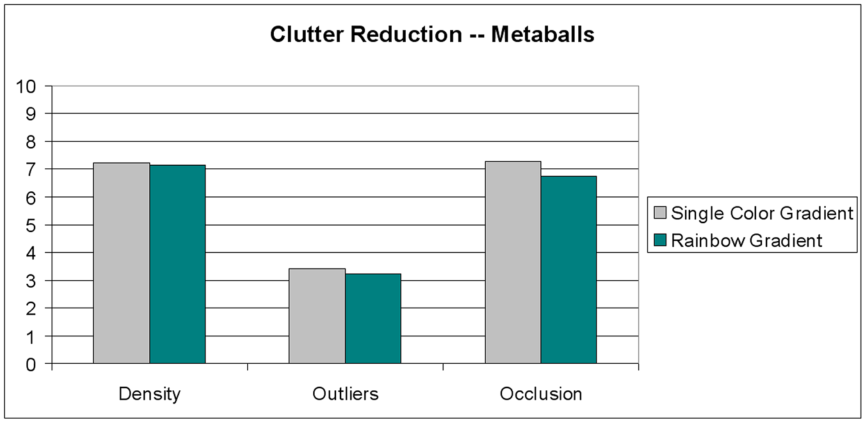


Figure 22: Metaballs: Effectiveness of clutter reduction.

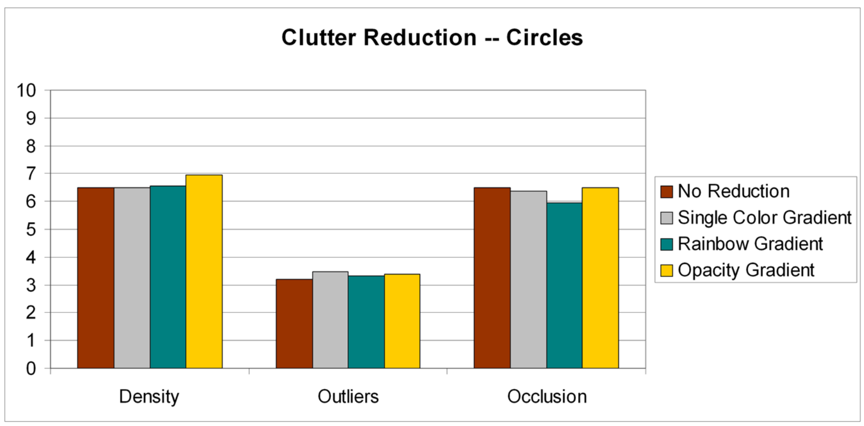


Figure 23: Circles: Effectiveness of clutter reduction.

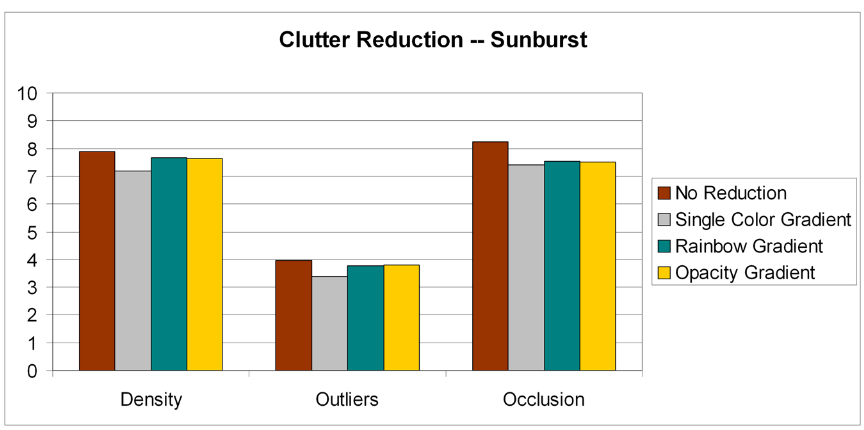


Figure 24: Sunburst: Effectiveness of clutter reduction.

For this section of the survey, users were asked to rate the visualizations based on how effective they were for performing three tasks:

- Locating busy areas,
- Identifying traffic routes, and
- Measuring air traffic volume

The format for evaluation was similar to the first section, but now a rating of 10 means the visualization is ideal for performing a task and a rating of 1 means it is impossible to perform the task. The results of user responses are summarized in Figures 25-27.

We can see that all visualizations were perceived as good for locating busy areas and measuring traffic volume, but no visualization was very good for identifying traffic routes. This is an expected result because the users were shown static images of the data. In reality, the users would be able to view the data animated over time and tasks such as identifying traffic routes would become more accessible.

The users also felt that clutter reduction techniques aided their ability to perform all three tasks. Although clutter reduction techniques were not always perceived as actually reducing clutter, they do help users perform tasks on the visualizations by focusing their attention on important information.

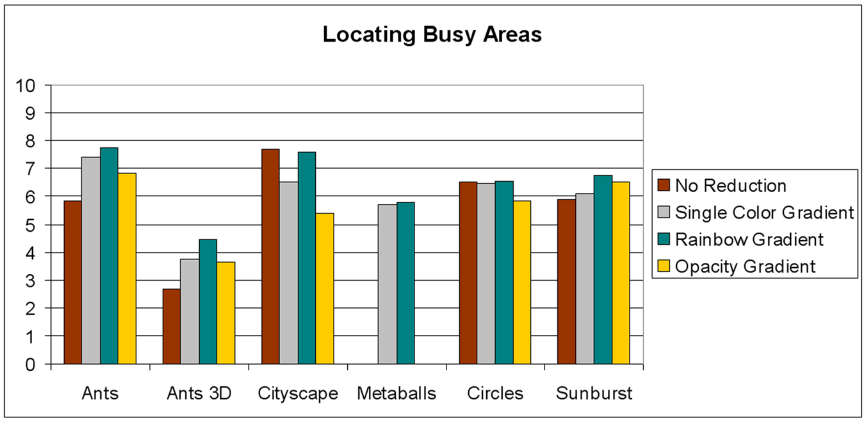


Figure 25: Task-Based Evaluation: Locating busy areas.

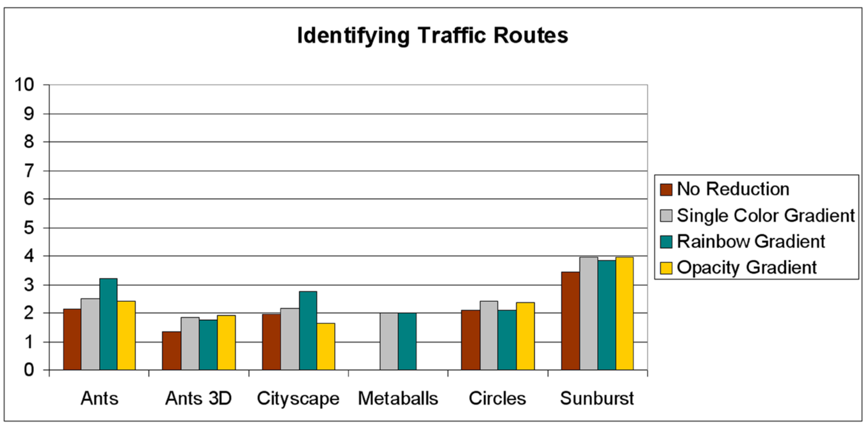


Figure 26: Task-Based Evaluation: Identifying traffic routes.

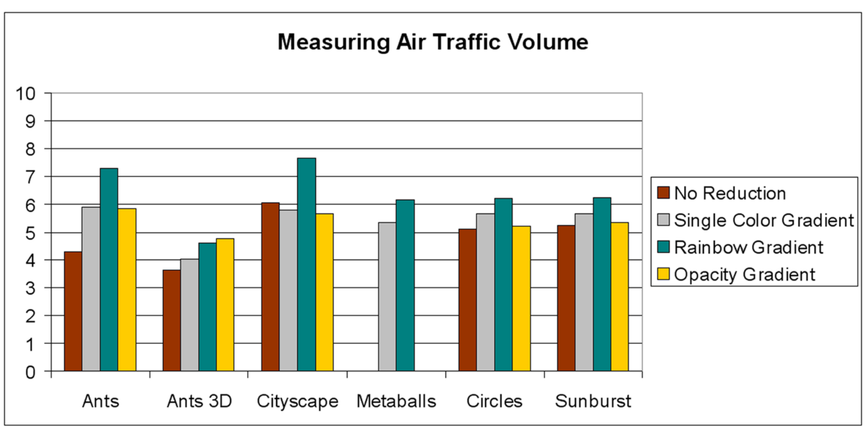


Figure 27: Task-Based Evaluation: Measuring air traffic volume.

7 Conclusions

Based on the work that was conducted for this project, it has become apparent that clutter measurement is an open task that has potential for many different applications. Although we only focused on the measurement of density, outliers, and occlusion, there are other possible measures that can also be applied more generally to other visualizations. The measurement strategies need to be tweaked for each application, but the concepts remain the same and can be used for many visualizations.

In addition, based on the results of user evaluation, we see that measurement strategies need to be adjusted to take more factors into consideration. For instance, the density measure of the Metaballs visualization should be adjusted to correspond with user perception of density. This can be accomplished by taking into consideration the area covered by the combined metaballs surface and looking at the color brightness over this area, rather than just counting the number of metaballs that make up the surface.

We have also seen that the effectiveness of any one clutter reduction technique depends somewhat on the tasks being performed. If we look at the different mappings in Figures 25 and 26, it is clear that Ants and Cityscape are better for locating busy areas, while Sunburst is better for identifying traffic routes. Here, user interaction plays an important role because it allows the user to select tools that are most effective for the task he or she is trying to perform.

Although there are no universal clutter measurement and reduction techniques, the methods discussed in this document can definitely be applied to many datasets other than air traffic, which is discussed in more detail in chapter 8.

8 Further Work

The work that has been completed for this project is only the tip of the iceberg for more detailed work that could be done, given more time.

The most pressing issue that was not addressed in this project due to time constraints is interactive user evaluation. The web survey that was used provided some good insight into how users perceive the visualizations, but it was missing the temporal aspect of the data set. The visualizations are meant to be explored interactively and viewed as a progression over time. It would be interesting to see how the animation/temporal aspect affects users' perception of the visualizations. More importantly, interaction would affect how users perform tasks and could have a significant effect on the perceived quality of all visualizations.

An important aspect of this project was to develop quantitative quality measures for visualizations. We focused on three metrics (density, outliers, and occlusion), but it is definitely possible to expand to more metrics. For instance, the amount of color present could be another metric. If more information was provided about the data set, such as points of origin, destinations, delay times, and passenger loads, then more tasks and therefore more metrics would likely become apparent. Similarly, other mappings could be developed with more information that could cater to the new tasks made available by more information. There are also many clutter reduction techniques that could be implemented. For instance, sampling could provide interesting information about the data set. With more information, many more possibilities would be available.

Finally, the work contained in this project could be applied to other data sets. The clutter measurement and reduction techniques discussed in this document could easily apply to other spatial and temporal data. We had an opportunity to explore a dataset of bubble movement in liquid while working on this project. This dataset contained the positions of several bubbles over some period of time and it fit naturally into the framework we were developing.

In addition to spatial and temporal data, our techniques could be extended to other types of data sets. None of our metrics or clutter reduction techniques relied on the temporal aspect of our data set directly, so they can easily be applied to non-temporal data. Furthermore, the specific metrics we explored were selected in order to provide the most general clutter measurements. Density, outliers, and occlusion are aspects of many, if not all, visualizations, even non-spatial ones. With some adjustments, they can be easily integrated for any type of data set or visualization. For instance, parallel coordinate visualizations can have dense areas, contain outliers, and have lines occluding other lines. Glyphs have the same problems. In some cases they can be arranged to not occlude each other, but density and outliers are still possible sources of clutter.

While some of our clutter reduction techniques were specific to our data set (i.e. Altitude Layers and all remapping techniques), many can easily be applied to other data sets, with or without a spatial aspect. For instance, the color and opacity gradients are general techniques that can be used for any kind of information display. What the gradients represent may be different, but the results will be the same. Camera angle optimization can also be applied to non-spatial data, though for this technique to be useful, the data would need to be displayed in 3D. The general concept of layers, or filtering, can be applied to filter any information from any data set. Because of the flexibility of the clutter reduction techniques discussed in this document, it is possible to extend this work into many different fields and many different visualizations.

References

[1] Airspace Concept Evaluation System (ACES). http://vams.arc.nasa.gov/activities/aces.html.

[2] Almir Olivette Artero and Maria Cristina Ferreira de Oliveira. Viz3d: effective exploratory visualization of large multidimensional data sets. *Proceedings of the 17th Brazilian Symposium on Computer Graphics and Image Processing*, page 340, 2004.

[3] Richard Brath. Metrics for effective information visualization. *Proceedings of the IEEE Symposium on Information Visualization*, page 108, 1997.

[4] Mei C. Chuah and Steven F. Roth. On the semantics of interactive visualizations. *Proceedings of the IEEE Symposium on Information Visualization*, page 29, 1996.

[5] Alan Dix and Geoffrey Ellis. By chance – enhancing interaction with large data sets through statistical sampling. *Proceedings of Advanced Visual Interfaces*, page 167, 2002.

[6] Stephen G. Eick. Aspects of network visualization. *Computer Graphics and Applications, IEEE*, 16(2):69, 1996.

[7] Lawrence E. Fraley. On verbal behavior: the fourth of four parts. *Behaviorology Today*, 8(2):3, 2005.

[8] Ying-Huey Fua, Matthew O. Ward, and Elke A. Rundensteiner. Hierarchical parallel coordinates for exploration of large datasets. *Proceedings of the IEEE Conference on Visualization*, page 43, 1999.

[9] Ulf Grenander and Anuj Srivastava. Probability models for clutter in natural images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(4):424, 2001.

[10] Simon Haykin, Wolfgang Stehwien, Cong Deng, Peter Weber, and Richard Mann. Classification of radar clutter in an air traffic control environment. *Proceedings of the IEEE*, 79(6):742, 1991.

[11] Stacie L. Hibino. A task-oriented view of information visualization. *CHI Extended Abstracts on Human Factors in Computing Systems*, page 178, 1999.

[12] Y.K. Leung and M.D. Apperley. A review and taxonomy of distortion oriented presentation techniques. *ACM Transactions on Computer-Human Interaction*, page 126, 1994.

[13] Nancy Miller, Beth Hetzler, Grant Nakamura, and Paul Whitney. The need for metrics in visual information analysis. *Proceedings of the Workshop on New Paradigms in Information Visualization and Manipulation*, page 24, 1997.

[14] Greg Mori. Recognizing objects in adversarial clutter. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, page 134, 2003.

[15] Ruth Rosenholtz, Yuanzhen Li, Jonathan Mansfield, and Zhenlan Jin. Feature congestion: a measure of display clutter. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, page 761, 2005.

[16] Steven F. Roth and Joe Mattis. Data characterization for intelligent graphics presentation. *Proceedings of the Conference on Human Factors in Computing Systems*, page 193, 1990.

[17] Joseph D. Rumbo. Consumer resistance in a world of advertising clutter: the case of adbusters. *Psychology and Marketing*, 19(2):127, 2002.

[18] M. Sheelagh, T. Carpendale, David J. Cowperthwaite, and F. David Fracchia. Distortion viewing techniques for 3-dimensional data. *Proceedings of the IEEE Symposium on Information Visualization*, page 46, 1996.

[19] Rebecca R. Springmeyer, Meera M. Blattner, and Nelson L. Max. A characterization of the scientific data analysis process. *Proceedings of the IEEE Conference on Visualization*, page 235, 1992.

[20] Lloyd A. Treinish. Task-specific visualization design: A case study in operational weather forecasting. *Proceedings of the IEEE Conference on Visualization*, page 405, 1998.

[21] Edward R. Tufte. *The Visual Display of Quantitative Information.* Graphics Press, Cheshire, CT, 1983.

[22] Edward R. Tufte. *Envisioning Information.* Graphics Press, Cheshire, CT, 1990.

[23] Edward R. Tufte. *Visual Explanations.* Graphics Press, Cheshire, CT, 1997.

[24] Stephen Wehrend and Clayton Lewis. A problem-oriented classification of visualization techniques. *Proceedings of the IEEE Conference on Visualization*, page 139, 1990.

[25] Marco A. Winckler, Philippe Palanque, and Carla M.D.S. Freitas. Tasks and scenario-based evaluation of information visualization techniques. *Proceedings of the ACM Conference on Task Models and Diagrams*, page 165, 2004.

[26] Pak Chung Wong and R. Daniel Bergeron. Multiresolution multidimensional wavelet brushing. *Proceedings of the IEEE Conference on Visualization*, page 141, 1996.

[27] Allison Woodruff, James Landay, and Michael Stonebraker. Constant density visualizations of non-uniform distributions of data. *Proceedings of the 11th annual ACM symposium on user interface software and technology*, page 19, 1998.

[28] Jing Yang, Wei Peng, Matthew O. Ward, and Elke A. Rundensteiner. Interactive hierarchical dimension ordering, spacing and filtering for exploration of high dimensional datasets. *Proceedings of the IEEE Symposium on Information Visualization*, page 105, 2003.

[29] Julie Yang-Pelez and Woodie C. Flowers. Information content measures of visual displays. *Proceedings of the IEEE Symposium on Information Visualization*, page 99, 2000.