Root Polar Layout of Internet Address Data for Security Administration

Glenn A. Fink* and Chris North†
Dept. of Computer Science, Virginia Polytechnic Institute and State University
Blacksburg, Virginia, USA 24061
http://infovis.cs.vt.edu

ABSTRACT

This paper introduces an adaptation of polar coordinates called "root polar plotting" that we have developed for our network pixel map—a computer security visualization capable of representing tens of thousands of hosts at a time. Root polar coordinates overcome two important problems of normal polar coordinates: plot density distortion and severe occlusion near the origin. We discuss several approaches we took while investigating this problem and provide empirical data from experiments we conducted comparing root polar coordinates against both normal polar and Cartesian coordinates. In any application where a polar plot would be useful but distortion of the data must be avoided, or where it is important to avoid some markers from being occluded by others, root polar coordinates may be useful.

Our approach provides: (1) a novel adaptation of polar coordinates that overcomes plotting distortion, (2) a means of plotting network data in near real-time without complex layout optimization, (3) an algorithm that reduces occlusion of plotted points while maintaining consistent placement, and (4) an empirical comparison of Cartesian vs. polar plots.

CR Categories: C.2.0 [General]: Security and protection—Software; K.6.m [Miscellaneous]: Security—Security visualizations; H.5.2 [User Interfaces]: Graphical user interfaces (GUI)—Visualizations; I.3.6 [Methodology and Techniques]: Graphics data structures and data types—Polar Coordinates;

Keywords: Internet Protocol, root polar plot, overlap, occlusion, plot density, pixel-oriented visualizations

1 PROBLEM STATEMENT

We are building the basis for a large-scale visualization of Internet Protocol (IP) addresses for computer security. Part of that effort involves plotting all the hosts recently involved in communication with a set of "home" network hosts. We especially wish to make communications between external and internal network hosts visible so that an administrator or security officer may quickly find unusual communication patterns such as high fan-out, high fan-in, and normality/abnormality of connections.

We want to display tens of thousands of IP addresses using pixel-oriented visualization techniques. Each IP version 4 address is thirty-two bits long and is written as w.x.y.z where w, x, y, and z are eight-bit unsigned integers called octets. Our first attempt was to plot each IP address in Cartesian coordinates using the first two octets of the address as the abscissa and the other two as the ordinate. While this method allowed users to relatively easily locate the general position of a given IP address on the plot, it did not lend itself to the idea that some of the hosts are locally administered, home hosts. Home was wherever your IP address mapped to on the plot. Using polar coordinates, it is more natural to locate home in the center of the plot. In interviews with our user community [4] and a recent usability study of a prototype security awareness tool [5] we learned that system administrators see the world of machines as falling into two basic trust categories, "us" and "them." [1] These trust categories may be broken down into any number of arbitrary categories from most to least trusted. With polar coordinates, a natural representation of trust levels would be nested circles with the most trusted level at the center.

Unfortunately, we encountered two major problems with normal polar coordinates: distortion and severe occlusion. This paper describes our efforts to overcome these problems and make the best possible use of the display space.

1.1 Assumptions

In our application of information visualization techniques to computer security, we seek to represent tens of thousands of hosts simultaneously and allow the user to display communication patterns between arbitrary locations. In this part of the application (the network pixel map), we do not concern ourselves with the communications between hosts, only about their relative placement on the plot. We assume only that we know an IP address (whether real or spoofed) for the host and that we can determine the relationship of a host to the home network based on this address and on recent experience with traffic from the host. In this section, we discuss the attributes of this data and our assumptions about the problem space.

First, we assume the existence of a known set of hosts that can be thought of as the "home" set; that is, machines that the user believes are benign and well-managed. We assume nothing about the size of the home set except that there are probably fewer hosts in the home set than outside it. Although the majority of traffic observed on a given network is local-only, over time the majority of the IP addresses seen will likely be from outside the home set.

Second, we assume data is collected within the home set and is biased toward it. Network data is gathered using packet "sniffers," hosts that collect traffic on the network. Especially in wired networks, the traffic visible to a single sniffer is highly localized. Data on other network segments cannot be sniffed without port mirroring or other techniques. For this stage of our study, we assume only a few sniffers provide the data we will visualize, and all the sniffers are located within the home set. This biases the data collected by limiting it to traffic originating on the home network, destined for it, or passing through it. Thus we would expect not to see traffic from one external host to another, but we do expect to see external to internal and internal-only traffic. These kinds of traffic patterns fit the polar layout more naturally than the Cartesian.

Third, we assume communication seen within the home set is mostly internal. From the perspective of the home set, there are

*e-mail: finkga@vt.edu
†e-mail:north@cs.vt.edu

three kinds of traffic:

1. Internal↔Internal: Traffic whose source and destination are within the home set
2. Internal↔External: Communication between home and non-home hosts (regardless of the originator)
3. External↔External: Traffic from outside the network bound for another external location but passing through the home network infrastructure.

Internal↔Internal traffic is is usually the majority observed at any interior point in the home set. While this kind of traffic has security significance, it is only of secondary importance to our study. Internal↔External traffic is of highest interest because we are seeking to show communication patterns that are most likely to have security implications. External↔External traffic is usually considered irrelevant since it typically comes no closer than a border router. We use a polar layout with the home set near the origin to help highlight Internal↔External and Internal↔Internal traffic flows.

Finally, we assume that network security analysts prefer to examine network data as close to real-time as possible. As a result, the presentation of data for our application must not rely on time-consuming preprocessing or computationally expensive space-optimized layout. Instead, we use the known characteristics of a host (*e.g.*, IP address, trust category, *etc*.) to place it directly on the plot. In case of collision with an already plotted host marker, we move the new marker to a nearby empty space. Once a marker is plotted, it will stay where it was placed until traffic to and from it disappears from the network.

1.2 Problems with Polar Plots

In many cases, a polar mapping of data may be more suitable than a Cartesian plot. Whenever there is a single point of reference and the notion of relevance to that point, polar coordinates may be useful. Examples of this kind of application are query results from a search engine, relevance rankings of documents in a collection to a given document, and social networks. However, polar coordinate plots have several serious problems including distortion, occlusion, and reduced area.

Problem 1: Distortion in polar plots. A casual glance at Figure 1 will show how the normal polar plot tends to distort the data by compacting it near the center and spreading it out at the edges. This characteristic has been useful in certain applications such as retinal emulation for robotic vision [10], but for our application the distortion destroys the picture we are trying to show. We wish to plot the home set of IP addresses at the center of the plot, but we also want to accentuate the individual members of the home set. Although the home set is much smaller than the set of external addresses, we want to give the home set equal space.

The distortion due to polar coordinates can be quantified. Uniformly spaced (n points at intervals of $\frac{2^{32}-1}{n-1}$) unsigned integer data plotted in a rectilinear plot has a constant plotting density. That is, the variance of the density, σ_D^2, approaches zero as the regularity of the grid approaches perfect uniformity. However, scaling the data to fit it on a 1000×1000 pixel screen introduces round-off error that will never allow us to attain perfect uniformity.

In contrast to the near uniformity of a Cartesian plot, a polar plot of uniformly spaced data is visibly more dense near the center and more sparse near the periphery. Thus, σ_D^2 is two to three orders of magnitude larger for polar plots than for Cartesian. However, we were surprised to find that σ_D^2 for root polar plots was only one order of magnitude greater. To refine our observations, we conducted a study using our network pixel map displaying the same set of uniformly spaced IP addresses on Cartesian, polar, and root polar coordinates. We then compared several plots of the same data, measuring the density variance of each graph. We found σ_D^2 for the normal polar plot was consistently higher, by an order of magnitude or more, than the same data plotted on either a Cartesian or root polar plot with the same area.

Problem 2: Occlusion in polar plots. The central clustering of polar coordinates causes a second problem: occlusion. With large numbers of points, a polar plot may have many layers of markers at the center, most of them hidden by more recently plotted markers. Occlusion is a general problem that happens in all kinds of graphs, but the problem is exacerbated in polar coordinates near the origin. We can fix occlusion by detecting collisions and moving the markers around, but this makes the perceived density near the center of the polar plot appear worse (see the normal polar plots in Figure 1). With collision resolution, the "solid" central area looks twice the diameter of the same plot without collision resolution. This shows how much overplotting is happening in a normal polar plot.

Problem 3: Smaller area of polar plots. One intrinsic problem with polar coordinates plots is that they have only $\frac{\pi}{4}$ the area of a square Cartesian plot whose sides are the length of the polar plot's diameter. We can mitigate the effect of the smaller plot space for our application by placing some markers (for the least trusted hosts) in the otherwise empty corner areas. Not all applications can make sensible use of the corner spaces, so they are either constrained to a smaller area or are subject to being clipped. In our study, to make the comparison between polar and Cartesian coordinates more clear, we have elected to compare our polar plots to Cartesian plots of identical area.

Although we handicap Cartesian coordinates by reducing their allowable area to that of a polar plot, Cartesian plots still have potentially higher capacity because displays are all rectangular arrays of square pixels. However, the difference is small enough that we did not think further handicapping Cartesian coordinates by setting the capacities equal to that of a polar plot would make a significant difference.

2 Potential Solutions

2.1 Solution 1: Fixing the Distortion of Polar Plots

After first discovering how useful polar plots could be for our application, our next discovery was how badly they distorted the plotting density of the data we were trying to display. In this section, we examine some of the ways we tried to overcome our problems with polar coordinates and how we finally arrived at the root polar approach.

2.1.1 Log and Exponential Polar Plots

Our first attempt at fixing the artificial central clustering was to convert the polar coordinates (ρ, θ) to $(\ln(\rho+1), \theta)$ in an attempt to smooth the distribution of the points. Log-polar sampling of image data is used in computer vision to reduce the amount of data that must be processed by a robot in real time. The log-polar transformation samples heavily near the focal point and very little at the periphery. This sampling approach mimics human and animal vision

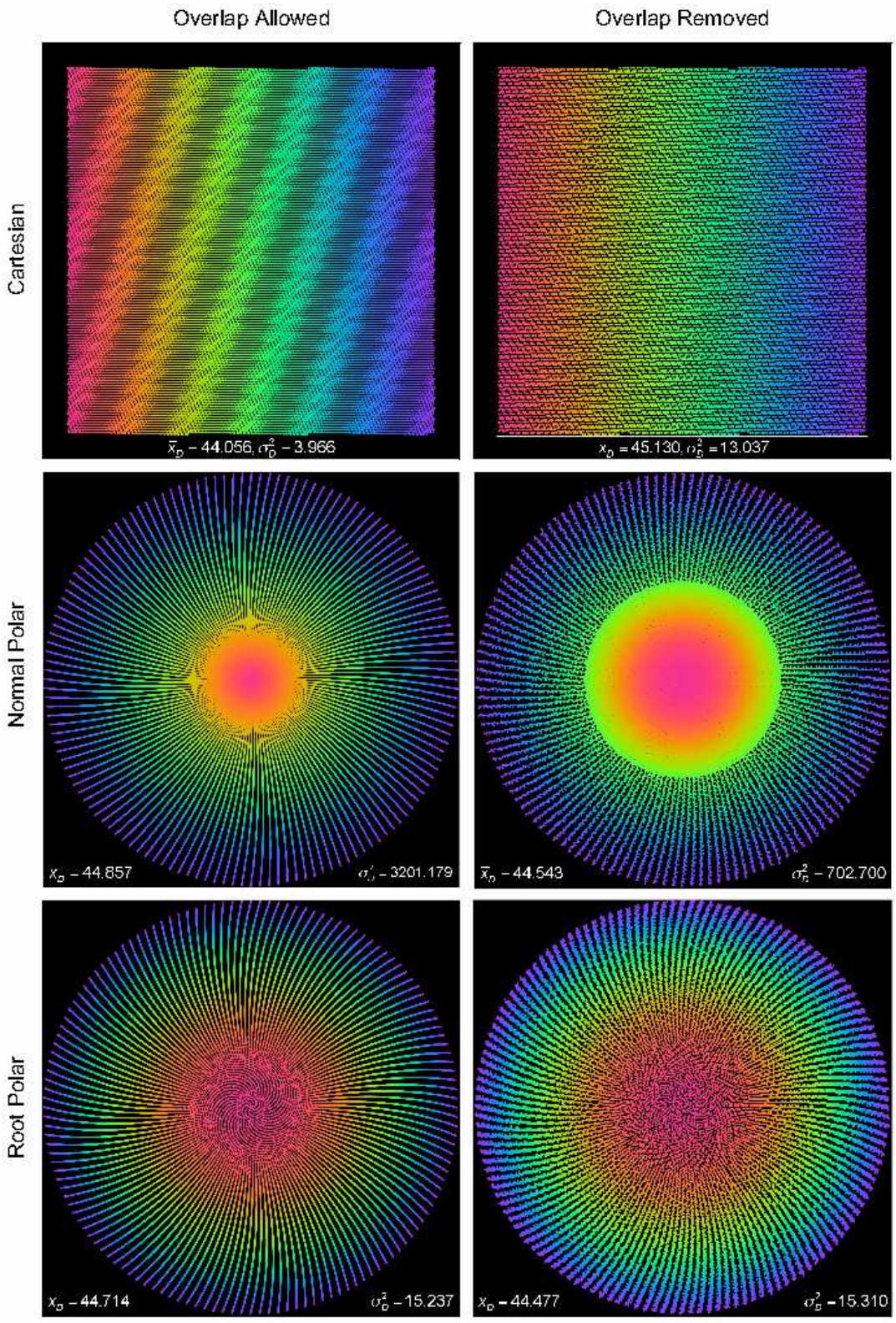

Figure 1: A series of plots of identical uniformly spaced IP address data. We color the markers using a Keim's HSI rainbow gradient [6] with lower numbers at the red end of the spectrum and the higher numbers toward the violet end. Plots on the right allow points to overlap. Those on the left use collision resolution to fix any overlap. The first two plots use Cartesian coordinates, the second two use normal polar coordinates and the third set are root polar plots. Note the large differences in the variance of the plotting density (σ_D^2) between the graph types.

and is most useful in active vision where the point of reference is constantly changing [2]. Log-Polar mapping is also used in digital imagery to create more secure watermarks of copyrighted images that are resistant to image transformations [8]. However, log-polar plotting exacerbated our original problem, squeezing all the points into a space less than half ($(\ln 2.0)^2 = 0.48$) the original plot area (see Figure 2).

Realizing that we had inadvertently accomplished the opposite of our intent, we tried the inverse, converting (ρ, θ) to $(\exp(\rho - 1), \theta)$. This pushed all the points out of the center leaving a hole. After experimenting with several different logarithmic and exponential bases, we became convinced that these two classes of functions, however they were modified, would never do what we intended. However, we note that the hole left in the center by the exponential plot could be used to an advantage later. We could inset another (non-exponential) polar plot in the center representing only the home IP address space.

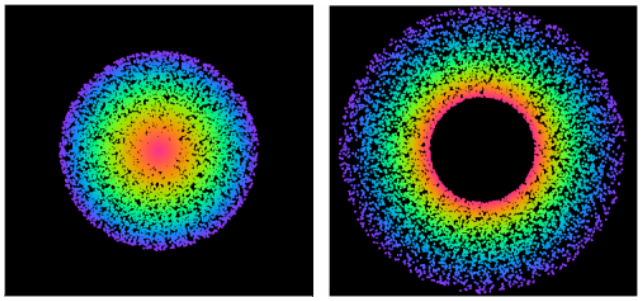

Figure 2: Polar plots using coordinates $(\ln(\rho + 1), \theta)$ and $(\exp(\rho - 1), \theta)$. The logarithmic plot exacerbates the central density distortion, while the exponential plot leaves a hole in the middle. Using different logarithmic bases does not correct the uneven density distortion.

2.1.2 Adaptive Density Polar Plots

Our second attempt involved predicting the optimal density of the plot *a priori* and adaptively adjusting the plotting density in the hopes of matching the optimal density. Given n markers of radius r and a plot radius of R we compute the optimal density as $\frac{nr^2}{R^2}$. We sort the IP addresses in ascending order so we can plot the central ones first. This allows us to sample the density of the already plotted area only once since no later data is placed closer to the center than what was already plotted. We then plot points on a horizon ring expanding from the origin. We calculate the plotting density at each ring by dividing the number of points plotted by the area inside the horizon ring. We then adjust the plotting density to approximate the optimal density.

Our adaptive algorithm determines the target polar coordinates of a marker, (ρ, θ), and adjusts ρ to be at least the current distance of the horizon from the origin. At each new horizon ring, we plot points near their target θ's until the density approaches the optimal density. In case of a collision with a plotted point, our algorithm adjusts θ by a small value, ε, calculated at each ring. We iteratively add ε to θ at each subsequent collision until either an empty space is found or we have adjusted θ by greater than 2π radians (360 degrees). If the algorithm does not find a noncolliding space after going all the way around, it adjusts both ρ and the horizon by the diameter of a marker and tries again. To decrease the density of the plot we increase the size of ε, essentially trying less hard to fill each ring. Conversely, to increase the plotting density, we decrease ε, making the algorithm try more places to fit a marker in the horizon ring before expanding it.

The adaptive density approach produced plots that were much less congested at the center than normal polar coordinates, but we found it very hard to adjust the density on the fly in a stable manner. Usually the algorithm would plot too densely near the center and then at about 90% of the plot radius it would have to adjust drastically, leaving the rest of the space only sparsely filled. The result was an uneven density graph interspersed with very sparse rings that made it appear concentric. We found that the adaptive density algorithm was inefficient because of the repeated density estimations required. The collision resolution approach often required many plotting attempts before an empty space was found. Finally, we found it very difficult to stabilize the density adjustment even for uniform random data. Real IP data, with its discontinuities was practically impossible to plot well using the adaptive algorithm. Although improvements in the algorithm may somewhat mitigate the problems, we abandoned the approach in favor of a more natural solution, root polar coordinates.

2.1.3 Root Polar Plots

What we needed was a function of ρ that grew more rapidly than linear at first and smoothly slowed its growth rate to the edge of the plot (at $\rho = 1$). A simple function that behaves this way is the square root function (see Figure 3). The early rapid growth of $f(\rho) = \sqrt{\rho}$ causes the points near the center to be spread out more. The later slow growth (relative to linear) causes the point density to increase near the periphery. This shape naturally counteracts the density distortion that arises from plotting in polar coordinates.

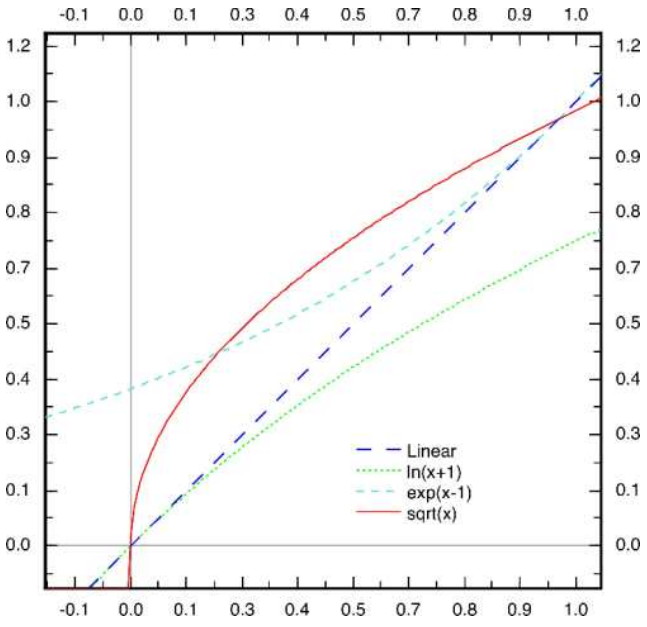

Figure 3: Graph of x, $\ln(x+1)$, $\exp(x-1)$, $\sqrt{x}$, and x showing their relative behaviors in the interval [0,1]. Note how $\sqrt{x}$ crosses x at $x = 0$, grows rapidly at first (causing relative sparseness near the center), and gradually tapers off until the plots meet again at $x = 1$. The growth pattern naturally counteracts the density distortion of normal polar coordinates in this interval.

Since we could constrain ρ to be between zero and one, we did not have to worry about taking square roots of negative numbers.

The resulting "root polar" coordinates worked very well both at removing the artificial clustering at the center and smoothing out the variance of the distribution of markers throughout the plot. Root polar plots are also not greatly affected by the plotting order, removing the reliance on *a priori* knowledge of the data or pre-sorting. To precisely measure the distortion differences, we used uniformly spaced data. On a rectilinear plot, this data looks like a set of regular, slanted lines, while the polar version produces a spiral. Any deviation from the regularity of the Cartesian plot is thus easy to notice and measure. It is also relatively easy to informally assess the regularity of the distance between the lines. With the root polar plot, it is easy to see that the spiraling lines are more regularly spaced over the length of the lines than a normal polar plot of the same data (see Figure 4).

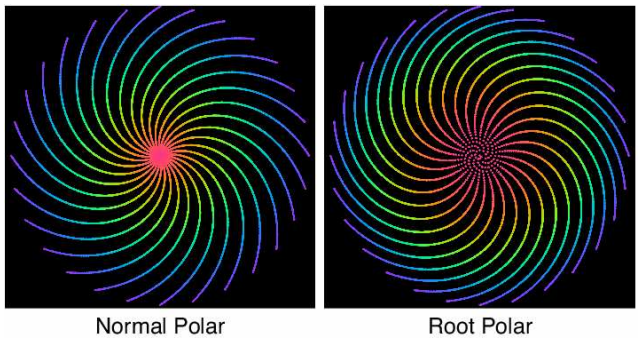

Figure 4: Normal and root polar plots of 5,000 uniformly spaced IP addresses with a marker radius of 2.5 pixels. Note the regularity of the distance between the spiral lines in the root polar plot over the length of the lines. This produces very low density variance. Note also the relative sparseness and reversed spiral direction at the center of the root polar plot. This indicates that the root polar plot does spatially distort the data near the center.

The root polar plot is not without its problems though. As ρ approaches zero, the slope of $\sqrt{\rho}$ approaches infinity. This leaves us with a very sparse center, but the result does not appreciably affect the density distribution of the rest of the points on the graph. A related feature arising from the steep slope of $\sqrt{\rho}$ near $\rho = 0.0$ causes the central spirals to reverse direction $\rho \leq 0.25$, where the slope is ≥ 1. The reversal implies that root polar plots induce some spatial distortion. Thus, polar plots reduce density distortion at the cost of distorting the relative positions of markers near the center. One can see this by comparing a normal and root polar plots of uniform data (see Figure 4).

2.2 Solution 2: Fixing occlusion in polar graphs

There are two basic approaches to resolving occlusion, grid and non-grid based approaches. In both approaches, if a new marker plots to a position overlapping existing markers we move it elsewhere. Using a grid simplifies the matter by constraining all positions to a finite number of non-overlapping slots. The grid simplifies the calculation of the new position at a potential cost of not packing markers quite as tightly as a non-grid approach might. For our purposes we consider only grid-based approaches.

Alternative 1: Nearest neighbor. The simplest way to position a new point that would occlude one or more other points is to move it to a nearby, unused grid location. We calculate the target rank, $\dot{\rho} = \left\lfloor \frac{\rho}{2r} + 0.5 \right\rfloor$. We keep track of which ranks we fill completely so we can select the lowest non-full rank greater than or equal to $\dot{\rho}$ as the actual rank, $\hat{\rho}$. Then we calculate the target slot, $\dot{\theta} = \left\lfloor \frac{\theta}{2\pi} \lfloor 2\pi\hat{\rho} \rfloor + 0.5 \right\rfloor$. If this slot is occupied, we try the slots to the left and right of $\dot{\theta}$, wrapping around at zero if needed. The first empty slot we find on rank $\hat{\rho}$ becomes the actual slot, $\hat{\theta}$.

In this application it is important to consider what moving the marker from $(\dot{\rho}, \dot{\theta})$ to the unoccupied position $(\hat{\rho}, \hat{\theta})$ will mean to the user. In our visualization we chose to place less trusted addresses further from the center (by increasing ρ), but the radial coordinate (θ) means nothing to us. So all the points at the same $\hat{\rho}$ are at the same level regardless of θ. Therefore, when a collision occurs, we resolve it by changing $\hat{\theta}$, increasing $\hat{\rho}$ only if there are no slots open at the rank $\dot{\rho}$.

We found that although the root polar plotting method works without regard to the plotting order, plotting the points sorted in nondecreasing ρ-order is helpful both to avoid collisions and to guarantee that our nearest-neighbor collision resolution algorithm will never place markers with larger ρ values closer to the origin than any marker with a smaller ρ value. If points are not plotted in nondecreasing ρ-order it is possible that collision resolution will cause some markers to be placed out of ρ-order as shown in Figure 5. The seriousness of this out-of-order plotting depends on how full the plot already is. The worst case happens when markers are plotted in nonincreasing ρ-order. Because the plotting order of recently seen IP addresses cannot be known in advance, it seems clear that collision resolution will result in some out-of-order plotting.

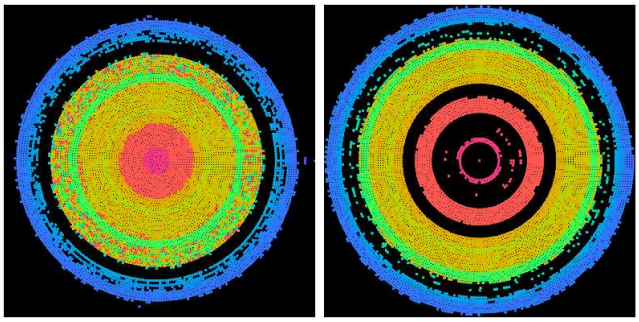

Figure 5: When IP addresses are plotted in random order collision resolution can force some out-of-order plotting. We plotted the same set of 8,513 real IP addresses with both root and normal coordinates in the same random order. Serious out-of-order plotting is evident in the normal polar plot (left) by the red points that appear far from the center, out of spectral (and thus numeric) order. The root polar plot (right) naturally spreads the data out better, so it is more robust with randomly ordered data.

Alternative 2: Space-filling curves. Keim mentions the use of space-filling curves [6], especially the Peano-Hilbert and the Morton curves, as an alternative to nearest neighbor placement. Space-filling curves are a method of mapping one-dimensional data into a two dimensional space. In this application a space-filling curve would plot the points along a single ρ line and then fold this ρ line into a regular two-dimensional, recursive pattern that fills the plot space and attempts to keep the ρ order of the points, within reason. Keim found these curves produced good clustering of the data but were difficult for users to interpret because the arrangement was not intuitive. We prefer a nearest-neighbor algorithm because it is not nearly so complex for dynamic layout as a space-filling curve would be.

Alternative 3: Keim's Gridfit. Keim goes on to describe his Gridfit algorithm [7] that optimizes the placement of all markers by hierarchically partitioning the data space into subregions. We

would like to try this method in future implementations and compare it with our current nearest-neighbor approach. Nearest neighbor is more straightforward to implement, is nearly as space efficient, and does not require *a priori* knowledge of the data as Gridfit does. We were also concerned about the potential complexity of using Gridfit in a near-realtime environment because of the potential for cascading movement of markers that might occur. For the meantime we chose to use the simplest method since the focus of this research is on the polar vs. Cartesian plots rather than on the packing algorithms.

2.3 Solution 3: Fixing the smaller area of polar plots

The most difficult intrinsic problem with polar coordinates is that given a square plotting area, a polar plot has only $\frac{\pi}{4}$ times as much usable space as as an equivalent Cartesian plot. An additional limitation is that a polar grid is a set of nested rings around the origin rather than tightly packed rows. Given the plot radius, R, and the radius of a marker, r, the number of enclosing ranks, k, in the plot is $\lceil \frac{R+r}{2r} \rceil - 1$. We have a single marker at the origin ($f(1) = 1$), and at the ith ring we have $\lfloor 2\pi i \rfloor$ slots. Thus the total number of markers in an area circumscribed by ring i is $f(i) = f(i-1) + \lfloor 2\pi i \rfloor$. Expanding this recursive formula yields $f(k) = \sum_{i=1}^{k} \lfloor 2\pi i \rfloor$ We were unable to derive a closed form for this sum, because of the floor function. However, we can provide a reasonably tight upper bound for it, $f(k) \leq \lfloor \pi k(k+1) \rfloor$.

In the circumscribed square whose sides are length $2R$ we can fit more markers, a full $4k^2$ of them. In our application, we can mitigate the smaller area of polar plots by allowing our polar plots to exceed their normal bounds but constraining the positioning of these points to the circumscribed rectangular region. Essentially, we can fill in the corners with the overflow. This approach allows us to approach the capacity of a Cartesian plot.

3 Empirical Comparison of Plot Types

We implemented a prototype network pixel map to compare Cartesian, polar, and root polar plots of IP addresses. We used a within-groups, full factorial design with four factors:

1. Plot type, 3 levels: Cartesian, polar, or root polar
2. Number of IP addresses (n), 6 levels: 1K, 2.5K, 5K, 10K, 25K, and 50K
3. Collision resolution (CR), 2 levels: on or off
4. Marker radius (r), 2 levels: 0.5 or 2.5 pixels

We ran all 72 possible iterations with uniformly spaced data rather than actual IP data as a control so we could measure differences in the variance of density, σ_D^2, without introducing artificial density distortions that occur in real data. The experimental conditions are summarized in Table 1.

When collision resolution is on, we constrain marker placement with a grid, and allow no two markers to occupy the same slot. The effect is to spread out the more densely populated areas, pushing the overflow into adjacent sparse areas. When collision resolution is off, we simply plot all markers at their true (rather than grid-constrained) coordinates. Marker radius interacts with collision resolution because larger markers take up more space and tend to cause more collisions. Large marker radius dramatically reduces the capacity of the plot area from about 455K to 18K nonoverlapping markers.

Group	Exp. Set	CR	r	n
I	1–3	Off	0.5	1K
	4–6	Off	0.5	2.5K
	7–9	Off	0.5	5K
	10–12	Off	0.5	10K
	13–15	Off	0.5	25K
	16–18	Off	0.5	50K
II	19–21	Off	2.5	1K
	22–24	Off	2.5	2.5K
	25–27	Off	2.5	5K
	28–30	Off	2.5	10K
	31–33	Off	2.5	25K
	34–36	Off	2.5	50K
III	37–39	On	0.5	1K
	40–42	On	0.5	2.5K
	43–45	On	0.5	5K
	46–48	On	0.5	10K
	49–51	On	0.5	25K
	52–54	On	0.5	50K
IV	55–57	On	2.5	1K
	58–60	On	2.5	2.5K
	61–63	On	2.5	5K
	64–66	On	2.5	10K
	67–69	On	2.5	25K
	70–72	On	2.5	50K

Table 1: Experimental design table. Each group (I-IV) of experiment sets holds the collision resolution and marker size constant. Comparisons between groups are not valid.

3.1 Comparison of Plotting Density

The most important difference between normal and root polar plots is in the distortion each induces on the data. The goal for the root polar plot was to avoid the σ_D^2 distortion seen in normal polar plots. We were not concerned with the θ-distortion that occurs near the center of root polar plots because θ is immaterial for our application. Since our experiments used uniform data the plotting density of a Cartesian plot is approximately uniform. Thus, the variance of the plotting density for the Cartesian plots, $\sigma_{D,Cartesian}^2$, approaches zero.

To estimate the density, we take a number of samples by placing a grid of constant-sized (relative to the marker radius), nonoverlapping tiles over the whole plot area (see 6). We then count the number of markers that overlapped each tile and estimate the population variance of the number of markers found over all the samples.

We found the mean density of all three plot types to be very close for each data set since we kept the area of all plots the same, and because the same amount of data was plotted each time. We observed the biggest difference in the comparison of the variances of density (see Figure 7). In every experiment set except 67-69 and 70-72, the computed variance of the normal polar plot was at least an order of magnitude higher than the variances of the corresponding Cartesian and root polar plots.

Experiment sets 67-69 and 70-72 were pathological cases with collision resolution, a large number of markers, and a large marker radius. Thus, these plots were filled beyond capacity and forced to be completely regular within the entire sampled plot area. Within the sampled area, the normal and root polar plots were constrained to be equal. In every other case,$\sigma_{D,normal}^2$ was between 2 and 269 times greater than $\sigma_{D,root}^2$. Except for the two cases with the largest number of markers (50K) and the small marker size (0.5), $\sigma_{D,root}^2$

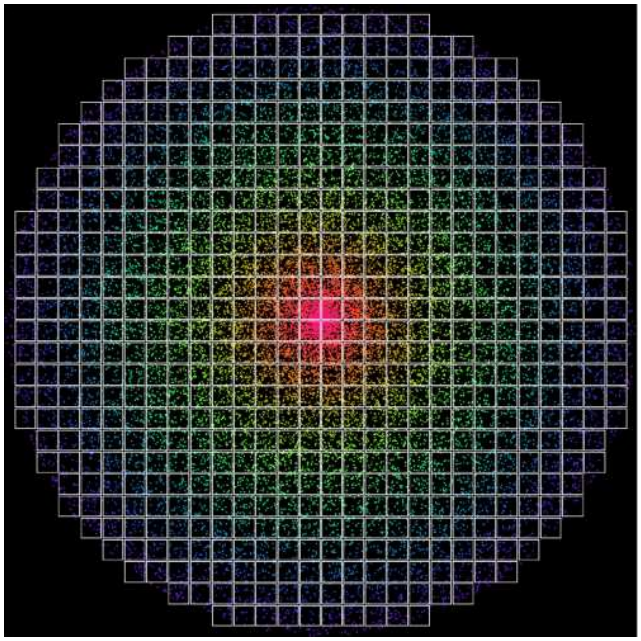

Figure 6: An example of how we sample density on the completed plots. Each white rectangle represents a density sample. We count the number of markers within each rectangle and compute the variance. We use this metric to compare density distortion between plot types.

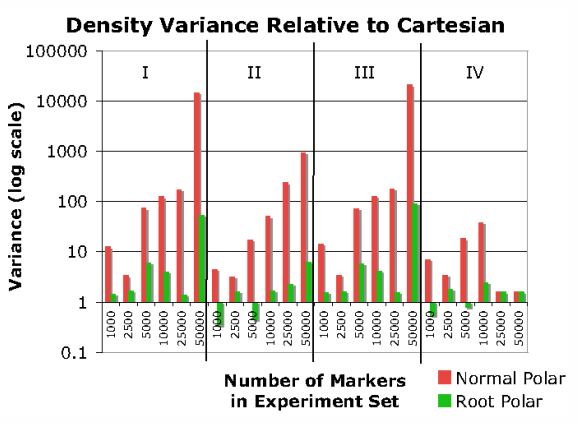

Figure 7: Ratios of plotting density variance for normal and root polar plots to that of Cartesian plots of identical data. Groups I and II have collision resolution disabled; groups I and III use the small marker size. Note that normal polar coordinates has more variation in every case except the the last two experiments. These are pathological cases with collision resolution and large numbers of large-sized markers.

was within one order of magnitude of $\sigma^2_{D,Cartesian}$. In these two exceptional cases, $\sigma^2_{D,root}$ was within two orders of magnitude of $\sigma^2_{D,Cartesian}$ while $\sigma^2_{D,normal}$ was five orders of magnitude greater.

Because we used uniform data with points at equal intervals, our study does not make use of any random process. Thus, statistical comparisons are not appropriate, and we can simply declare that our study numerically proves that root polar coordinates distort the plotting density less than normal polar coordinates do for the tested conditions. A previous study used uniform random data and showed the same results statistically. We believe our findings are a strong indication that root polar coordinates will distort the plotting density less than normal polar coordinates under all conditions likely to occur in our application.

3.2 Comparison of Collision Rates

We define a "collision" to be whenever the marker of a data point to be plotted overlaps one or more already plotted markers. The mean collision rate is the expected probability that a new point on a plot will at least partially occlude some other points. We count collisions each time the algorithm attempts to place a new marker where one is already plotted.

All three plot types use the same collision resolution algorithm, so we expect the differences in mean collision rate to be all due to the plotting densities. We found only one experiment set where the mean collision rate for normal polar coordinates was less than that of root polar coordinates. All other times the collision rate of normal polar coordinates was one to four orders of magnitude higher than either root polar or Cartesian.

The variance of the collision rate tells how much the mean collision rate is expected to change from place to place on the plot. To calculate the variance of the collision rate, we kept running totals of the number of collisions in each rank of each plot type. Then we calculated the variance of the number of collisions in each rank. Normal polar coordinates distort the data so much that the collision variance of a normal polar plot is considerably higher than for Cartesian or root polar plots. Basically, variance of a collision rate indicates how well the data is spread around or clustered on a given plot. Using uniform data allows us to measure the distortion induced by each kind of plot. We believe that the very high collision variances for normal polar coordinates are a second evidence of the σ^2_D distortion induced by polar coordinates.

3.3 Comparison of Run Times

The run time of the plots is a much less robust measure of effectiveness because run time (especially in scripting language implementations) is influenced by so many uncontrollable factors, such as garbage collection and external utilization of the processor. However, recording the run times helped us to see a correlation between collision rate and run time. When a plot becomes very densely packed and collision resolution is enabled, the plotting algorithms spend most of their time resolving collisions.

On the whole, both types of polar plots ran in about twice the time as the Cartesian plots of the same data under the same conditions. Normal polar plots outperformed root polar slightly, but this is not surprising since the two share the same code with the only difference being the $\sqrt{\rho}$ coordinate transform. In a few high-density cases the higher number of collisions incurred by the normal polar plots actually caused it to run slower than the equivalent root polar plots. The polar and Cartesian plotting algorithms are also very similar with only a few small implementation differences in the way the coordinates are calculated and the way collisions are resolved.

4 Application

Two questions remain about the application of root polar coordinates to IP address plotting for near real-time security monitoring:

1. Is the root polar plotting method equally apt with real IP address data?
2. How do we place the "home" network's addresses in the center?

In this section, we will answer these questions and discuss our implementation approach.

4.1 Real IP Data

We searched for good samples of IP address data on the Internet, but found no suitable sample. Available data had either been anonymized (destroying the original distribution) or used DNS names rather than IP addresses (and in most cases names had changed, making reverse lookup impossible). As a work-around, we collected 8,513 unique IP addresses from a single workstation using tcpdump over a period of two weeks. While the distribution of this data is almost certainly not the same as the true (and unknowable) distribution of all IP address data, it does fit our assumption that data collected by local sensors will be locally biased.

We found the distribution of IP addresses to be clustered into several bands (see Figure 8). Because of the banding we incorporated a spreading algorithm in the calculation of polar coordinates. The spreading algorithm does rely on *a priori* knowledge of the distribution of the data, but it is not unreasonable that this knowledge could be part of a local profile that changes only rarely.

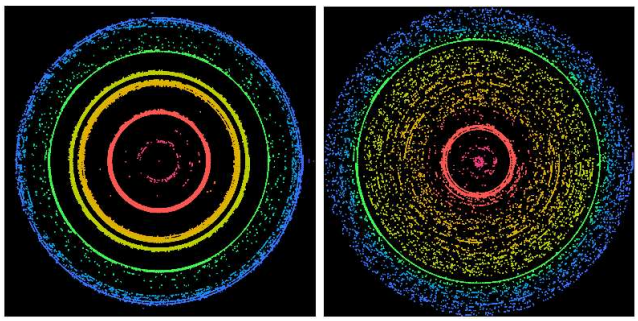

Figure 8: A root polar plots (with CR) of 8,513 IP addresses collected from a single point via tcpdump. Major clusters appear in the 24.0.0.0/8, 60.0.0.0/5, 80.0.0.0/5, and 128.173.0.0/16 blocks (the latter being the campus's IP block and colored bright green). Smaller clusters appear higher up as well. The plot on the left shows the raw data without spreading. On the right we have applied the spreading algorithm.

4.2 Putting Home in the Center

We have seen that root polar coordinates can effectively spread real IP data so that the display space is much more evenly utilized. However the real advantage of root polar over normal polar plots is clearest when we tackle the problem of keeping the "home" network at the center of the plot.

To do this, we have developed the concept of trust levels. Not all hosts on the Internet are trusted equally. In fact, most organizations have a "white list" of address spaces they administer and a "black list" of address ranges known for previous malicious behavior. Between the two is the murky set of addresses that are simply unknown. Of the unknown addresses, there are probably some that an administrator may expect to be commonly accessed by his users, such as search engines. An administrator within the organization may also subdivide the organization's "whitelist" into a set of machines he administers (and thus has a personal stake in) and another set with the rest of the enterprise's addresses. For our application, we define up to five trust levels: self, enterprise, safe, unknown, and dangerous. Of course an administrator may choose to use only the minimal set of trust levels: "us" and "them." The number of levels (as long as it stays reasonably small) is unimportant. Using trust levels, we can plot the most trusted ("home") nodes in the center to obtain plots like those shown in Figure 9.

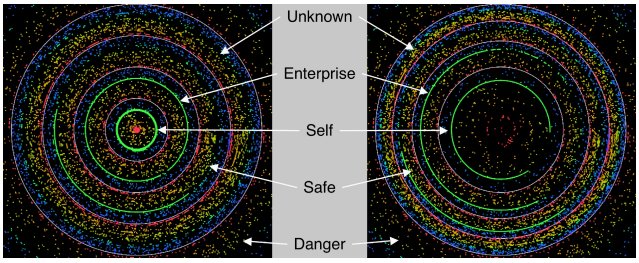

Figure 9: Normal and root polar plots of IP data with five arbitrary trust levels superimposed. As before, the color gradient indicates the relative value of the addresses plotted (spectral with lower values toward the red end and higher values toward the violet end). Now, however, we plot addresses from low to high value within their trust level. All addresses within a given trust band are assumed to be (dis)trusted equally.

Given l trust levels, we plot $l-1$ rings, with the ith ring at $\rho_i = \frac{Ri}{l-1}$ from the origin. For root polar plots, we transform ρ_i as well so the innermost circle is disproportionately large. Each IP address has an associated trust level assigned according to how its CIDR block is classified by administrative policy. To place an IP address whose trust level is t we calculate a trust modifier $m = \frac{t}{l-1}$. Then we plot the marker at $m\rho$ rather than ρ (where ρ has already been subjected to coordinate transforms). This places the marker inside the appropriate trust ring or outside all of them (for "dangerous" hosts).

In Figure 9, the center (representing machines managed by the user) is larger, allowing more accurate analysis of the machines the user cares about and natural aggregation of those he cares less about. An important difference between the plots may be seen in the green-colored markers in the innermost trust level. In the normal plot, the distortion coupled with collision resolution has forced these markers to circle about the origin several times. But in the root polar plot these hosts do not even complete a single revolution even though there are many of them.

Further, we can expand or contract the size of the central area by applying different powers of ρ instead of $\sqrt{\rho} = \rho^{0.5}$. Smaller powers of ρ expand the center and larger ones contract it. This usage is reminiscent of the "fisheye lens" [9] often used as a focus+context approach in information visualizations. However, we use root polar coordinates as a robust layout approach that minimizes occlusion over the entire graph rather than simply to magnify a particular focal area.

In another study of root polar coordinates, we determined that 0.5 is the optimal root of ρ for reducing σ_D^2 distortion as compared to Cartesian plots. However, using other powers of ρ may be useful to adjust the focus during security monitoring work. Root polar plotting with trust levels provides a simple way to place the area of most interest to the user centrally while avoiding the distortion inherent in normal polar plots.

5 Future Work

This section discusses a few other important directions for our future research in this area.

Usability studies. One important future direction we plan to take with this research is to perform usability studies on polar *vs.* Cartesian coordinates to understand the cognitive implications of plot layout. Particularly, we wish to know whether users can find particular addresses and determine a host's trust level on polar plots as quickly and accurately as they can on Cartesian plots. Another area for usability studies is to determine how many markers a user can comprehend on a single plot. Our root polar plotting prototype works well for plots of 100,000 or more markers, but that does not mean that the plot is usable. Further studies are needed now that the technical groundwork has been laid.

Aggregation: Making good use of occlusion. In some cases occlusion is good. In fact, aggregation techniques are simply controlled occlusion. Estan *et al.*'s AutoFocus [3] aggregates network traffic into groups responsible for major amounts of observed communication. Their purpose is to make it simpler for humans to quickly comprehend who the most active communicators are. We would like to investigate using localized aggregation of machines that a user sees as a group to simplify the analyst's job.

Showing Communication Lines. The next step for the network pixel map is to put it in the larger context of end-to-end communications between hosts over the Internet (see Figure 10). We have outlined our vision for end-to-end communications visualization in other papers [1, 5]. The pixel map only provides a layout for the hosts that are observed communicating on the network. It cannot show communications between these hosts by itself. What we plan to do next is to place mirrored network pixel maps side by side and draw communication lines between them. This will enable users to see communication patterns such as fan-out and fan-in easily. From this concept, we have created a prototype of the network view in OpenGL that allows users to manipulate the display in 3D space (see Figure 11).

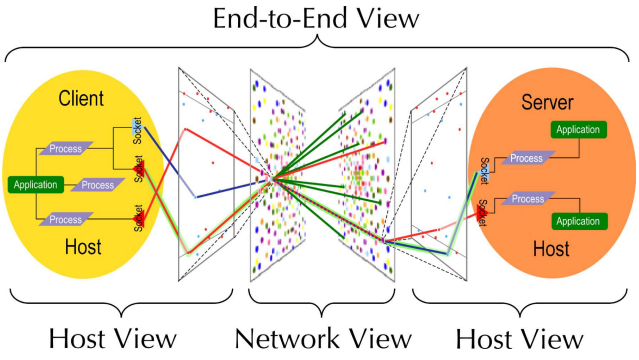

Figure 10: Overview of the Network Eye application.

Perceptual issues. A variety of perceptual issues need to be investigated to determine the best ways to help our users improve their awareness of network events. We plan to include a fisheye lens to aid in selection of the very small host markers. Without some kind of magnification, pixel-oriented techniques cannot easily be used interactively. We will enable users to zoom into the home area and examine local-to-local traffic as well, because a some proportion of security problems comes from malicious insiders. Finally,

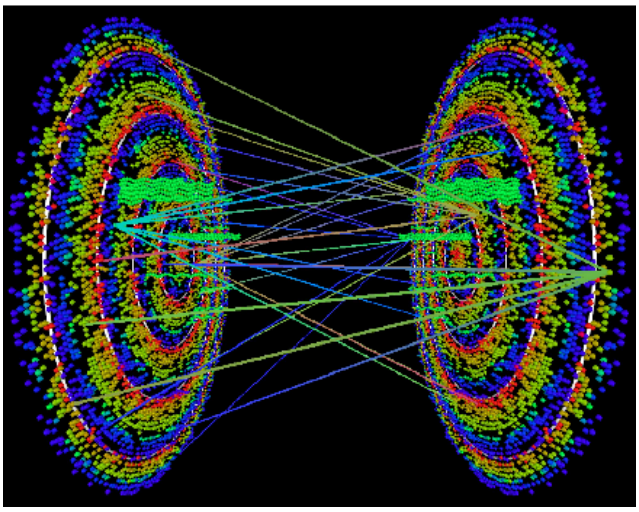

Figure 11: OpenGL prototype of the Network View.

we need to experiment with fading out inactive markers and replacing them with new markers. We need to ensure that our plots maintain spatial consistency even when hosts appear and disappear over time.

6 Conclusion

We have proposed root polar coordinates as an alternative to normal polar coordinates when distortion of the data's density must be minimized. This work offers the following contributions to the field:

- Provided a new layout that meets all the needs identified in the problem section. We have shown that root polar coordinates meets the needs outlined in our problem statement, namely that there are relatively few "home" hosts and a relatively large number of less trusted hosts, that the presentation may be dynamic, and that most communication seen from the inside of an organization is internal.
- Provided a way to overcome the plotting density distortion of normal polar coordinates We have demonstrated how root polar coordinates avoid the distortion inherent in normal polar coordinates. Particularly, square root polar coordinates neither concentrate a large number of points near the origin, nor do they spread points out near the periphery. We believe this shows that root polar coordinates are most useful when occlusion would garble the message of the data. In our application, using root polar coordinates will allow the users to clearly see the "home" hosts and will not press other categories of data into the home space.
- Provided a means of plotting data in near real time without complex optimization. Root polar plots help spread the data around naturally without having to resort to computationally expensive optimization methods. Our study has shown that fewer collisions and thus less work to resolve them results when root polar plotting is used as opposed to normal polar plotting. We have also shown that root polar plots with collision resolution are quite robust when data is plotted in random order. Thus, we have shown that root polar plotting is a good choice for presenting data in a polar layout under tight, near real-time constraints.

- Provided empirical comparison of Cartesian vs. polar vs. root polar plots. We have presented a numeric proof that the plotting density distortion of root polar coordinates is less than that of normal polar coordinates. We believe our conclusions will help visualization specialists to decide when to use normal polar, root polar, and Cartesian coordinates.

7 Acknowledgments

This research was supported in part by a National Science Foundation Integrated Graduate Education and Research Training (IGERT) grant (award DGE-9987586).

References

[1] Robert Ball, Glenn A. Fink, Anand Rathi, Sumit Shah, and Chris North. Home-centric visualization of network traffic for security administration. In *Proceedings of the 2004 ACM Workshop on Visualization and Data Mining for Computer Security*, International Conference on Information Visualization (IV2000), pages 55–64. ACM, ACM Press, 2004.

[2] A. Bernardino and J. Santos-Victor. Correlation based vergence control using log-polar images, 1996.

[3] Cristian Estan, Stefan Savage, and George Varghese. Automatically inferring patterns of resource consumption in network traffic. In *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 137–148. ACM Special Interest Group on Data Communication, ACM Press, 2003.

[4] Glenn A. Fink, Ricardo Correa, and Chris North. System administrators and their security awareness tools, 2005.

[5] Glenn A. Fink, Paul Muessig, and Chris North. Visual correlation of host processes and network traffic. In *Proceedings of VizSec 2005*. IEEE, October 2005.

[6] Daniel A. Keim. Designing pixel-oriented visualization techniques: theory and applications. *IEEE Transactions on Visualization and Computer Graphics*, 6(1):59–78, 2000.

[7] Daniel A. Keim and Annemarie Herrmann. The gridfit algorithm: An efficient and effective approach to visualizing large amounts of spatial data. In *Proceedings of the Conference on Visualization '98*, pages 181–188. IEEE Visualization, 1998.

[8] M. Kutter. Watermarking resisting to translation, rotation and scaling, 1998.

[9] Manojit Sarkar and Marc H. Brown. Graphical fisheye views of graphs. In *CHI '92: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 83–91, New York, NY, USA, 1992. ACM Press.

[10] M. Tistarelli and G. Sandini. On the advantage of polar and log-polar mapping for direct estimation of time-to-impact from optical flow. *PAMI*, 15(4):401–410, April 1993.