Probabilistic Geometry and Information Content (an Introduction to Corob Theory)

Version 1.1 Jan 2003

Douglas J. Matzke, PhD
Lawrence Technologies, LLC, [1]
matzke@ieee.org

P. N. Lawrence, PhD [1]
nick@LT.com

Abstract

The topic of probabilistic geometry has been researched in the field of mathematics for well over 50 years. Applying the intrinsic yet unintuitive metrics in these high-dimensional spaces to the information arena is conceptually very tricky. Pentti Kanerva found results related to this field in the mid 80s and applied them to content addressable memories. Dr. P. Nick Lawrence also rediscovered a more generalized version of these results in the early 90s and has developed it into his patented computational theory called *Corobs,* which stands for *Cor*relational-Algorithm *Ob*jects. Recently, the link between quantum theory and Corob Theory was researched under DOD SBIR funding. This paper/presentation gives an overview of this field including the key concepts of how to implement useful computing knowing that randomly chosen points are all an expected or *standard distance* apart (robust equidistance of sqrt(N/6) in a real unit N-cube as N>20). These ideas are particularly relevant to the ANPA audience because of the direct application to "Program Universe" goals and results, since in a binary unit N-cube the distance and discrimination metrics are both XOR. For a link to the slides for this presentation see the Yahoo anpa-list email archive.

1. Introduction

The primary idea beneath ANPA's combinatorial hierarchy goals is the hope that known stable constants and laws of physics spontaneously emerge due to the combinatorics of low-level random information processes. As a result of years of research at Lawrence Technologies, LLC on random neurological processes and more recently random quantum processes, we have developed a high-dimensional, computation model we call "Corob Theory". This paper introduces the key ideas of Corob theory, which are related to the obscure mathematical field of probabilistic geometry. The goal of corob theory is to understand and mimic our ability to see and do things like previously seen and done. Corob theory and program universe goals are similar because both are interested in how something stable can emerge from purely random processes. Even though Corob theory started as a high-dimensional neurological based computation model, it was recently expanded to include the high-dimensional complex-valued

[1] 5485 Belt Line Rd, Suite 200, Dallas, Texas 75254, USA, see http://www.LT.com

quantum corobs. These results depend on the intrinsic and emergent yet unintuitive distance metric properties of high-dimensional spaces.

1.1 Approach and Concepts

The Corob Technology is based on the idea that neurology somehow uses randomness to do useful similarity computing, and understanding this approach will lead to computers that act more like living systems, thereby performing like synthetic organisms or Synthorgs™. This thinking led to the rediscovery of the little known mathematical property from probabilistic geometry [1] that randomly chosen points within a high-dimensional unit N-cube approaches a growing expected or standard Cartesian distance $\sqrt{N/6}$ as the number of dimensions grows large. Even though these results are statistical in nature, the standard deviation is a constant $\sqrt{7/120}$ = 0.2415, so effectively the actual range of the standard distance becomes extremely narrow and approaches the expected mean standard distance with high levels of certainty. We call this tendency toward the expected *standard distance* the fundamental Corob theorem.

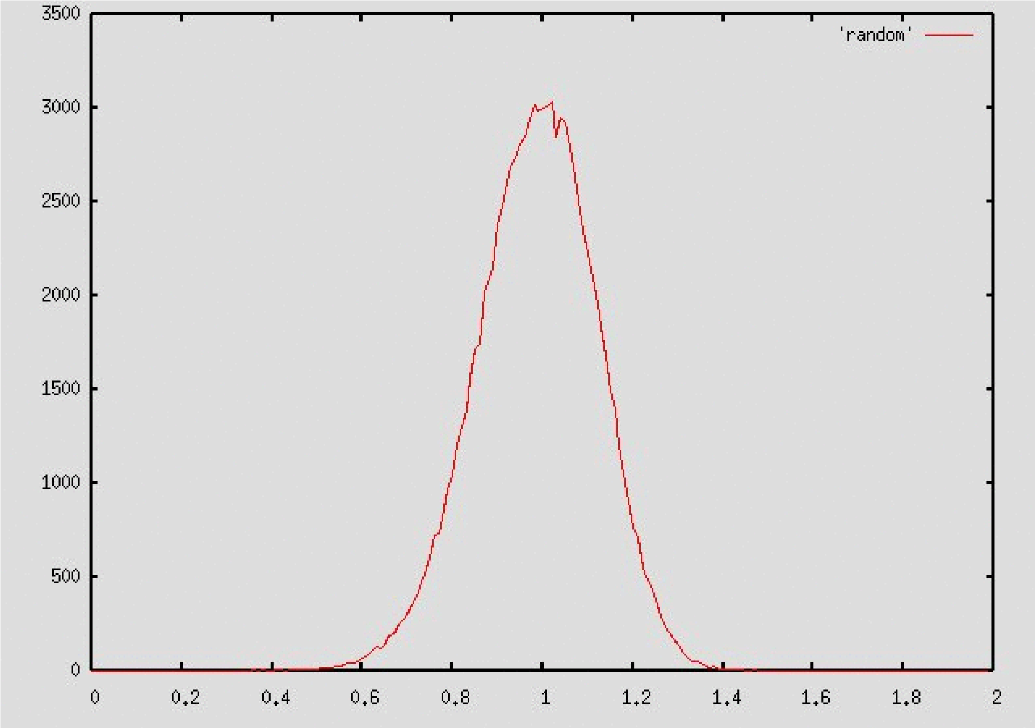

Figure 1. Random process with mean of 1.0

2. Randomness and Statistics

The results presented in this paper act like random processes with a mean and a standard deviation. Therefore, these two terms are introduced and discussed here. This section is a small refresher course on the key aspects of probability mathematics the corob theory relies on. Also, we define our own terms that rely on these statistical properties.

2.1 Mean or Expected Value

If a number of *uniformly distributed random values* are fed into *a random process* they produce a mean value with a high level of precision, so the random process tends to the mean, which represents the *expected value*. The random process actually generates values slightly smaller or larger than the mean, as shown in Figure 1. So even though the process tends to the expected value or mean, the actual values spread out around that center point. As one might expect random values close to the mean are more common than values further away. This spread can be characterized using the standard deviation and confidence interval, which are discussed next.

2.2 Standard Deviation or Dispersion

The exact amount of spread of the values from the random process is defined by the measure called *standard deviation*, which is the square root of the variance. Random processes naturally produce values closer to the mean value with a higher probability. The dispersion of values from the random process falls within the normalized interval defined by one *standard deviation* with a probability (or confidence interval) of 68.2%. This means that roughly two out of every three values falls within one standard deviation, nineteen out of every twenty values with within two standard deviations, etc (see Table 1). In contrast, it is possible but *extremely unlikely* that a random process will produce a value greater than 4 or 5 standard deviations away from the mean. This probability can be computed using the error function if the distribution is Gaussian. This relationship between probability and distance deviation from the mean has important information theoretic significance for the random processes discussed in this paper.

Table 1. Confidence Interval vs. standard deviation

Standard Deviations	Confidence Interval
±1	0.6826895
±2	0.9544997
±3	0.9973002
±4	0.9999366
±5	0.9999994

3. High Dimensional Spaces

The *primary random process* forming the basis of corob technology is simply applying the *Cartesian distance metric* between two sets of N uniformly distributed, bounded, random values, where each set represents the address of a random point inside a high-dimensional unit N-cube. This random process can be thought of in two ways, as a mean of the probabilistic process or an ensemble result similar to the concept of the center of gravity. The ensemble results and interpretation appears to be directly related to the probabilistic phenomena.

As this paper will demonstrate, the mean (and thereby the standard deviation) of several probabilistic Cartesian distances is related to the square root of the number of dimensions $\sim\sqrt{N}$, with some unexpected results as N grows much larger than the typical 3 Euclidean dimensions (N>>20). This chapter provides a step-by-step mathematical introduction to these main ideas.

3.1 Cartesian Distance Metric

The random process is simply the distance metric between any pair of points. Given an N dimensional unit N-cube and address values ($0 \le x_i \le 1$ and $0 \le y_i \le 1$) in each of the N dimensions for points X = [x_1, x_2, …, x_i, …, x_N] and Y = [y_1, y_2, …, y_i, …,y_N], then the Cartesian distance is simply the square root of the sum of differences squared between the two points or $d = \sqrt{\sum_i (x_i - y_i)^2} = \sqrt{\sum_i (z_i)^2}$. The distance to the origin in three dimensions (for X, Y and Z) is simply the well-known distance formula $d = \sqrt{(z_1)^2 + (z_2)^2 + (z_3)^2}$ or $d = \sqrt{(X)^2 + (Y)^2 + (Z)^2}$.

The primary unexpected result can now be discussed, which occurs when the values x_i and y_i are *randomly generated* with a uniform distribution random number generator, thereby creating a probabilistic geometry [1]. Under these conditions the distance between two randomly chosen points tends to an *expected constant standard distance* $\sim \sqrt{N/6}$ as the number of dimensions grows large N>>20. This equidistance tendency will be discussed next.

3.2 Tendency towards Equidistance

In three dimensions, the histogram of distances between randomly generated points X to the center point Y = [0.5, 0.5, …, 0.5] looks like the graph in Figure 2. This histogram is basically triangular, where the asymmetry is due to fewer occurrences of the largest distances since they are non-uniformly located only in the extreme corners of the n-cube.

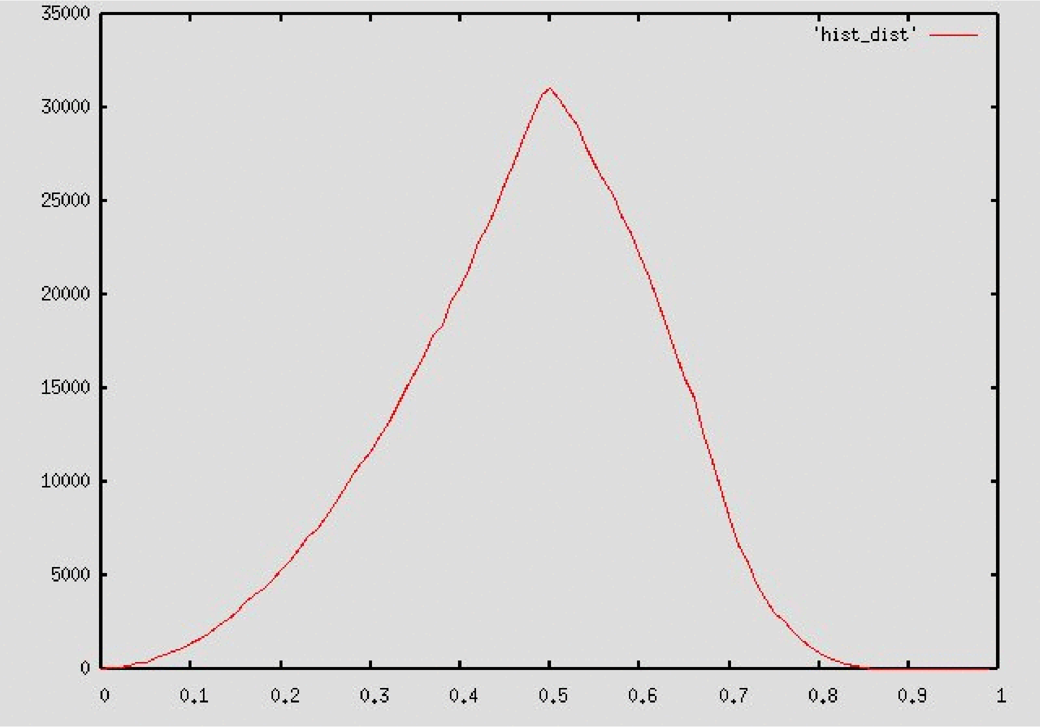

Figure 2. Distances for random points to center point in 3-space

The interesting thing to note is that Figure 2 is exactly the same random process as Figure 1 except that the number of dimensions representing each point is N = 3 rather than N = 12. The distances to the center point can be generalized for any number of dimensions and the result defines a probabilistic geometry, where any randomly chosen point probabilistically *tends* to be at the distance ~ $\sqrt{N/12}$ from the center of the space. This distance to the center is conceptually equivalent to an *expected standard radius* that grows with the $\sqrt{N}$ and the tendency toward that particular value grows very strong with increasing N because the *probabilistic distance metrics are dependent on* $\sqrt{N}$ but this *standard deviation is also a constant*!

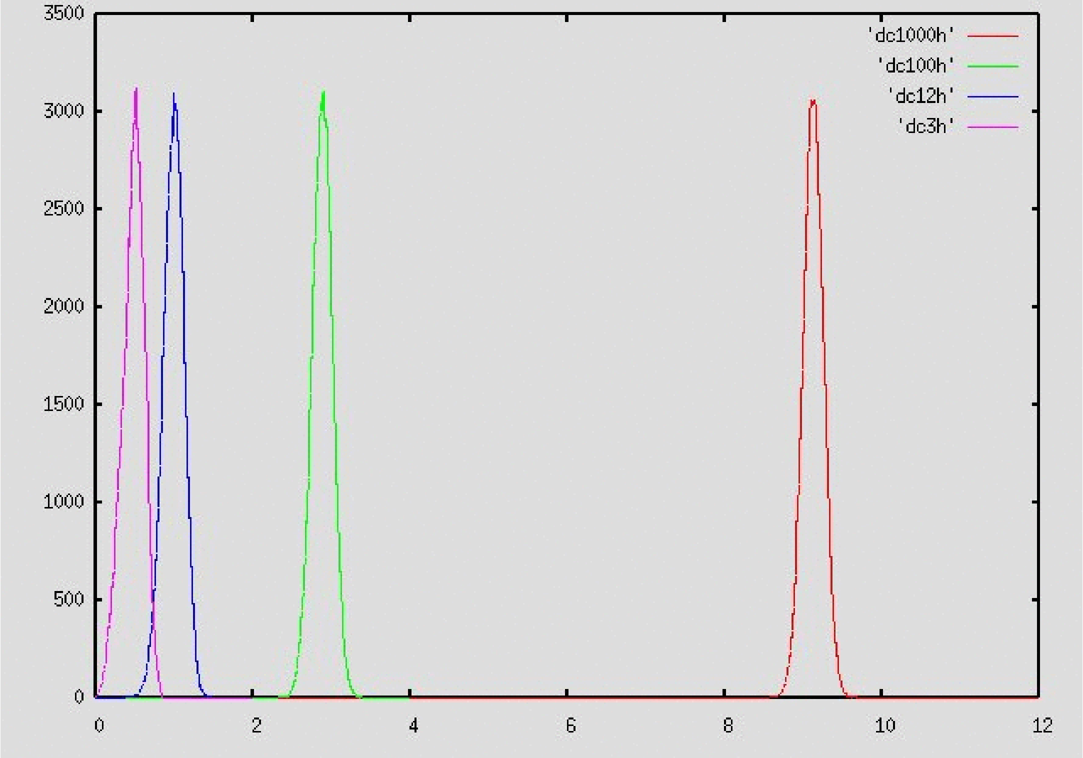

Figure 3. Histogram of Standard radius for N = 3, 12, 100 and 1000

Figure 3 illustrates the *standard radius* and dispersion of the values with increasing N. If the standard deviation is actually computed for each of the cases the standard deviation is the *constant value* $\sqrt{1/60}$. Since the standard deviation indicates the spread of values around the mean standard radius, the actual spread becomes a smaller and smaller percentage of the mean that grows with increasing N. Therefore, normalizing the standard deviation by dividing by the expected radius produces the ratio $\sqrt{1/60}/\sqrt{N/12}=\sqrt{1/60*12/N}=\sqrt{1/5N}$. The *normalized standard deviation actually shrinks* with N (and in the limit tends to 0 for very large N). This means the *standard radius tends strongly to a standard constant* value (for each N) with increasing N. See Figure 4, which is a normalized version of Figure 3. Because of this property, probabilistic geometries are easy to explore using simple programs with randomly chosen points, because they quickly converge to their respective mean values, and it is *high unlikely* to find a point that is 5-10 standard deviations away from the mean.

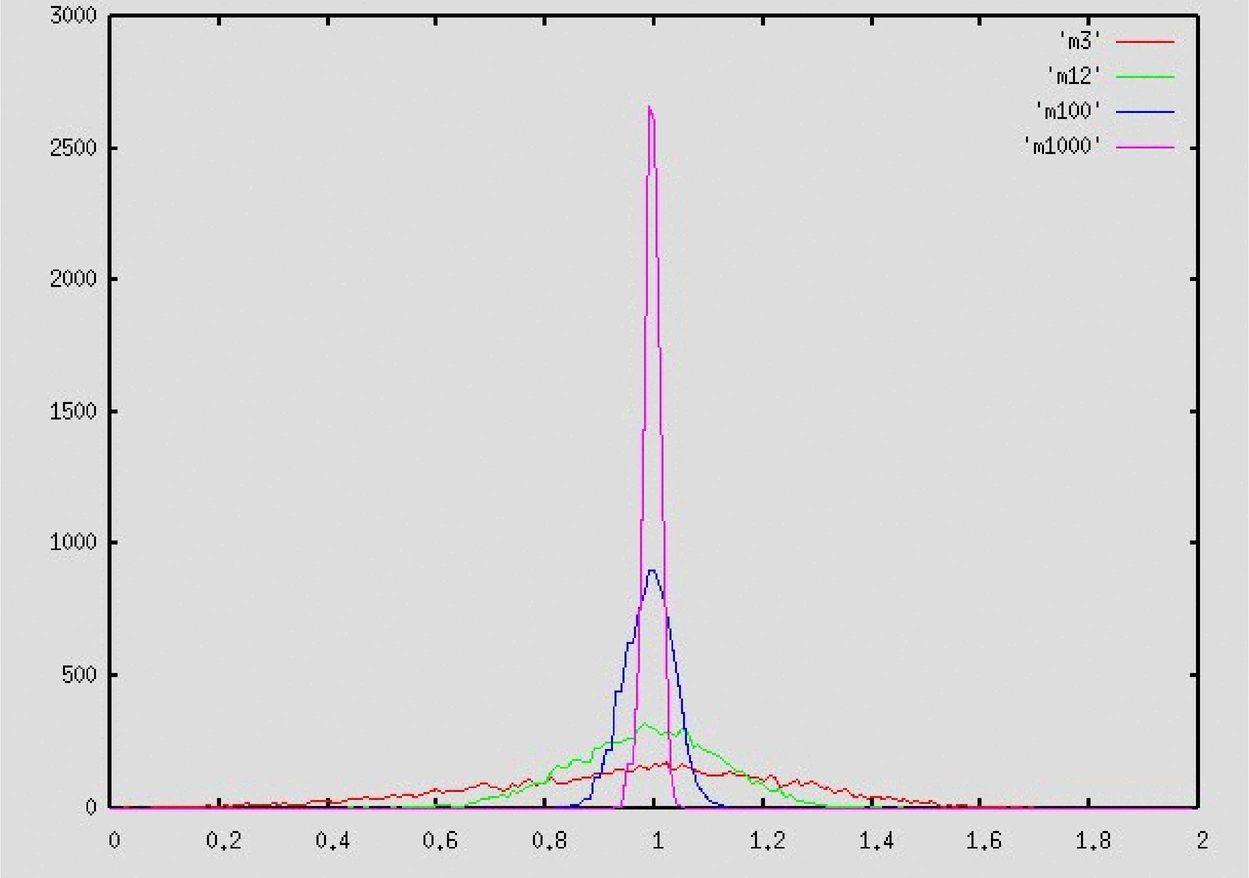

Figure 4. Normalized standard deviation of radius for N=3, 12, 100 and 1000

3.3 Probabilistic Geometries

The label of *standard radius* evokes the idea that the random points are located on the surface of a high-dimensional N-*sphere*. One way to attempt to confirm this is to find the distance between a 1000 randomly chosen points and the origin Y = [0, 0, …, 0] with N=1000 and the result is the distance $\sqrt{N/3}$, which happens to be twice the standard radius $2\sqrt{N/12} = \sqrt{4N/12} = \sqrt{N/3}$. This distance is unexpected because the known geometrical distance from the origin to the center is simply $d = \sqrt{\sum_i (0.5)^2} = \sqrt{N(0.25)} = \sqrt{N/4}$. Since the random point is actually the radius $\sqrt{N/12}$ away from the center, this defines a right triangle $\left(\sqrt{N/4}\right)^2 + \left(\sqrt{N/12}\right)^2 = \left(\sqrt{N/3}\right)^2$, which simplifies to N/4 + N/12 = 3N/12 + N/12= 4N/12 = N/3. Therefore the distance between any randomly chosen point to the origin is $\sqrt{N/3}$, but even more unexpected is that this is also the distance to *any random corner*!! Likewise, the standard deviation of this standard corner distance approaches 0 with increasing N. This result is hard to visualize because the distance between a random point and *every random corner* is the constant $\sqrt{N/3}$. This is similar to Pentti Kanerva's [2] distance between any two random corners of an n-cube $\sqrt{N/2}$ since the right triangle from the center to two random corners has the property $\left(\sqrt{N/4}\right)^2+\left(\sqrt{N/4}\right)^2=\left(\sqrt{N/2}\right)^2$ and has the binomial distribution that can be derived using Pascal's triangle.

The most relevant statistical metric is the distance between *two randomly chosen points* inside the unit N-cube tends to the *standard distance* $\sqrt{N/6}$, with the standard deviation approaching 0 as N grows to infinity. It is not that the other points do not exist; it is they are *extremely* hard to find using random processes. For a large number of random points, they all tend to a *standard radius* from the center, a *standard diameter* (D=2r) from any corner and a *standard distance* from each other. Points with these properties *do not form* an N-sphere inside an N-cube, but rather a high-dimensional *tetrahedron* or N-*equihedron* (or N-shell), since all the points tend to be equidistant. This idea is illustrated in Figure 5, where all the blue points are equidistance from the center red dot (and more importantly from each other). If any point (big yellow dot) was then moved to the center, the big red dot would be among the remaining blue points. These distance properties are definitely wrong for an N-sphere so we use the term N-shell or N-*equihedron*.

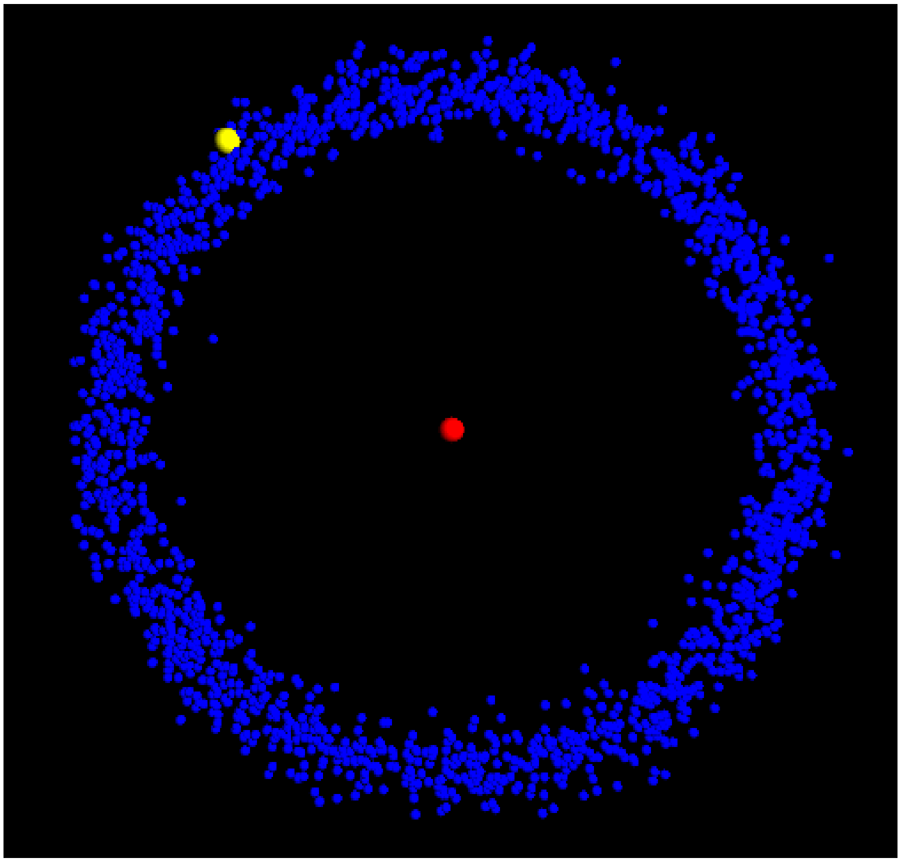

Figure 5. Planar Visualization of Equidistance Property

An alternative visualization of all the equidistance properties is shown in Figure 6, which depicts this topology using a planar equilateral triangle to represent the N-equihedron and all lines are distances normalized by the expected standard radius. Note that any lines of the same color are the same length even if it is impossible to draw them as such in a planar graphic.

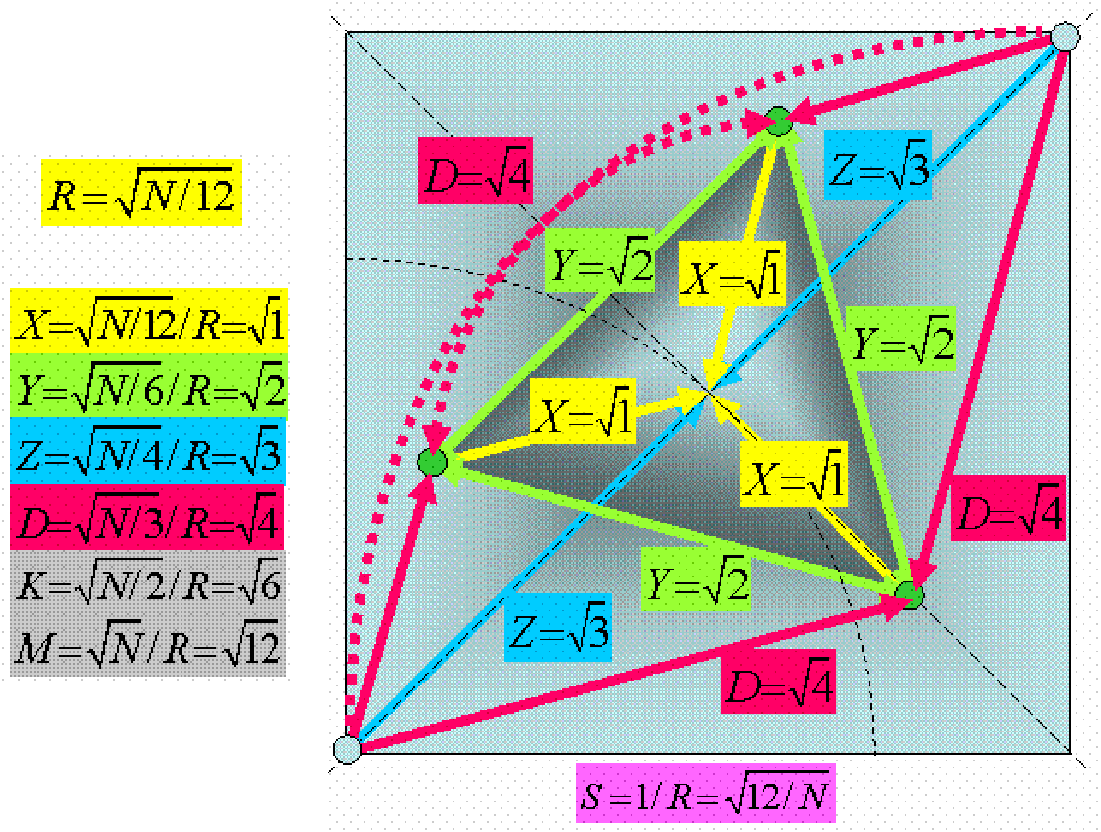

Figure 6. Standard distances from the perspective of green random points

An important fact regarding Figure 6 is all the distances have been normalized by dividing by the *expected standard radius* R resulting in *all standard distances becoming invariants or constants.* For example, with the normalized radius X = R/R ≈ 1, then similarly the expected standard distance is Y $\approx \sqrt{2}$, the normalized corner to center distance is Z $\approx \sqrt{3}$, the standard corner distance is D = 2R = 2 $\approx \sqrt{4}$ and the n-cube's major diagonal is M $\approx \sqrt{12}$. The normalized Kanerva distance is K $\approx \sqrt{6}$. This means all expected distances are true *constants* independent of N and with vanishing standard deviations for very large N. Table 2 illustrates these same metrics in a tabular fashion depending on the preferred perspective of unit side or unit radius.

Since the standard radius and major diagonal are constants for any number of N, then the normalized length of every side S must shrink with increasing N or S = 1/R $\approx \sqrt{12/N}$. Maintaining a unit radius forces the N-cube to shrink and maintaining a unit side length on the N-cube means the standard radius increases proportional to $\sqrt{N}$. Since 12 is the magic number used for both the radius and N-cube side, when the number of dimensions N = 12 then the lengths R = S = 1 = volume. When the radius R = 1 then this acts like the unitarity constraint from quantum mechanics. This simple set of mathematical relationships is rich with potential meaning regarding the size or volume of things in both the classical and quantum domain depending on the number of dimensions and the normalization perspective. The number 12 is a special constant that emerges from this analysis, where these two aspects are indistinguishable.

Table 2. Geometric and probabilistic properties of unit N-cubes

Line	Type of Property	Distance with Unit Side	Distance with Unit Radius	Description of distance metric
1	Geom	$= 1$	$\approx \sqrt{12/\mathrm{N}}$	Side: any corner to a nearest corner
2	Prob	$R \approx \sqrt{\mathrm{N}/12}$	≈ 1	Radius: midpoint to random point
3	Prob	$Y \approx \sqrt{\mathrm{N}/6}$	$\approx \sqrt{2}$	Random point to random point
4	Geom	$Z = \sqrt{\mathrm{N}/4}$	$\approx \sqrt{3}$	Any corner to midpoint
5	Prob	$D \approx \sqrt{\mathrm{N}/3}$	$\approx \sqrt{4}$	Random corner to random point
6	Prob	$K \approx \sqrt{\mathrm{N}/2}$	$\approx \sqrt{6}$	Random corner to random corner
7	Geom	$M = \sqrt{\mathrm{N}}$	$\approx \sqrt{12}$	Any corner to most distant corner

A final point about Figure 6 is the arc drawn at distance $\sqrt{3}$ from the origin. Obviously there are more points outside this arc then inside this arc, so this represents why the ensemble length $\sqrt{3}$ is greater than the geometrical corner-center distance of $\sqrt{4}$.

This unit N-cube space is a distance metric space but nothing has been said about the angles. From standard geometry it is know that for right triangles then $a^2 + b^2 = h^2$ where sides a and b form a right angle and h is the hypotenuse. This property happens to be true for the expected standard distance $\sqrt{1}^2 + \sqrt{1}^2 = \sqrt{2}^2$ and the Kanerva distance $\sqrt{3}^2 + \sqrt{3}^2 = \sqrt{6}^2$, which suggests all randomly chosen points (or corners) are orthogonal to each other when treated as vectors from the center of the space. Obviously, the angles shown between the yellow lines of the standard radiuses all approach 90° even though Figure 6 cannot be drawn to scale. Since the standard distance is probabilistic, the angle between them is also not exact. The angle between to two vectors (defined as midpoint to each random point) approaches an *expected standard angle* of 90° as N grows. Figure 7 shows how the standard deviation shrinks with increasing N such that for N>3000 the standard deviation of the standard angle is <1°. The angle is computed with the formula in Figure 7 using the inner product $\langle \boldsymbol{x} | \boldsymbol{y} \rangle = \sqrt{\sum x_i y_i}$ and norm $\|\boldsymbol{x}\| = \sqrt{\sum x_i^2}$ formulas.

$$q = \cos^{-1}\left(\frac{\langle \boldsymbol{x} | \boldsymbol{y} \rangle}{\|\boldsymbol{x}\| \; \|\boldsymbol{y}\|} \right)$$

Size of N	Standard Deviation	
	Inner Prod	As Angle
100	.1000	5.758°
1000	.0315	1.816°
10,000	.0100	0.563°

Figure 7. Expected Standard Angle approaches 90° for large N

3.4 Soft Tokens or Corobs

The entire probabilistic geometry is based on the premise that a set of random points generate these probabilities and at Lawrence Technologies we call such a random point a “corob” or soft token. A soft token is a valid term we have borrowed from communications theory, which has the identical meaning, where the tokens are maximally spread out through an N-space providing an *error-correcting region* around each token. It is also know from coding theory the maximum information density occurs when the tokens are uniformly spread throughout the space and they are equally spaced. These properties naturally occur for corobs and have been known about in the field of probabilistic geometries for decades. The primary invention of corob theory is how to coax useful Turing complete computing from these randomly populated metric N-spaces.

3.5 Computing with Soft Tokens

Computing with random corobs requires defining tokens and assigning particular symbolic meaning to each of the tokens. This is equivalent to assigning *meaning* to a particular random address in the subspace of dimension N. All the other random points are extremely far way because the standard deviation of distance metrics are proportional to $\sqrt{1/N}$. Computation is simply mapping values from the environment into points in a subspace (using a sensor), finding which points are closest to *meaningful* known points, converting to another subspace (lobe), and finally converting the soft tokens back into the tradition numeric or string literals (actor). Corob computing is simply the process of creating spaces with the correct distance relationships to perform your desired behavior by moving the computation state around within the subspaces. This process can be shown to be Turing complete because lobes can define arbitrary logic gates and also represent dynamically stored state similar to a latch or cross-coupled NAND-gates. The following three sections discuss the various spatial and temporal relationship properties.

3.6 N-spaces and Subspaces

If an N-dimensional space is split in two, both of the resulting subspaces have the same properties regarding random tokens, as the whole N-space but with lower fidelity due to the increase in the normalized standard deviation. This is equivalent to the holographic property and can be easily understood as a noisy version of the original token. In general, random corobs are a set of random address values defined for a subspace of particular size N. Subspaces are concatenated together and sampled which effects tokens relative addresses depending on the desired operation. Remember that the directions up, down, right, left, east and west etc do not exist in subspaces. The only useful metric is similarity of distances compared to randomly chosen points, which were previously produced or known about in the system. For this reason, synthetic organisms respond in ways similar to previously seen, known or educated actions.

3.7 Cardinal, Ordinal and Continuous Relationships

Most of computing involves defining relationships between tokens, and corob theory geometrically represents all such relationships in N-spaces. The three kinds of relationships are:

1) Cardinal or *unordered* tokens (such as a list of names, properties or colors).
2) Ordered tokens (such as the alphabet where letter “c” is closer to “d” than “m”)
3) Continuous spaces (such as 1-D for temperature and 2-D for images, etc)

Cardinal tokens are directly implemented using the random corobs already defined since they tend to be equally spaced with no apparent ordering. A subspace can hold M cardinal tokens without any confusion by choosing the size N of the subspace such that some minimum spacing is guaranteed (i.e. 5-10 standard deviations) for that number of tokens.

Ordered tokens require that the distance between tokens is proportional to the difference in their adjacent sequence number. The distances between adjacent tokens must be smaller than standard distance and can be constructed using the following procedure. First decide what percentage of standard distance the tokens should be spaced apart (i.e. use s = 30% but depends on the application). Second, generate a random point $c_{1=i}$ and assign that to the first token "a". Third, generate another random point $c_{2=i+1}$ and use it as a random angle reference point at standard distance d. Interpolate between the two points to find a point close to c1 by using the formula c_1*s + c_2(1-s) to compute the address of token "b". Repeat steps 2 & 3 between remaining points c_i and c_{i+1}. The resulting pearls on a string each have a distance metric that is smaller for its immediate token neighbor(s) than to the other tokens. When any token is compared to any of the pre-defined tokens, the closest tokens are easy to find. Also, the nearest neighbors are unique because it is extremely unlikely that random corobs will be less than the standard distance apart. This close spacing of sequential tokens is called a *string corob* (can also form a ring corob).

Continuous spaces are conceptually equivalent to ordered-tokens except that a continuous number of points must maintain the linear spacing as expected in a continuous variable. So if a continuous space is pre-populated with a set of ordered grid points, it is possible find an arbitrary number of interpolated points between each grid point. Unfortunately, the interpolated metric distances between grid points act as if they droop like fabric between tent poles because this process represents simulating a linear 1-D space within a high-dimensional N-subspace with unusual probabilistic metrics. Therefore, a construct was invented and patented called the *toothpick method* for string corobs, which effectively randomizes the drooping effect. The string corob can be adapted for representing 1-D, 2-D, etc continuous spaces as illustrated in Figure 8.

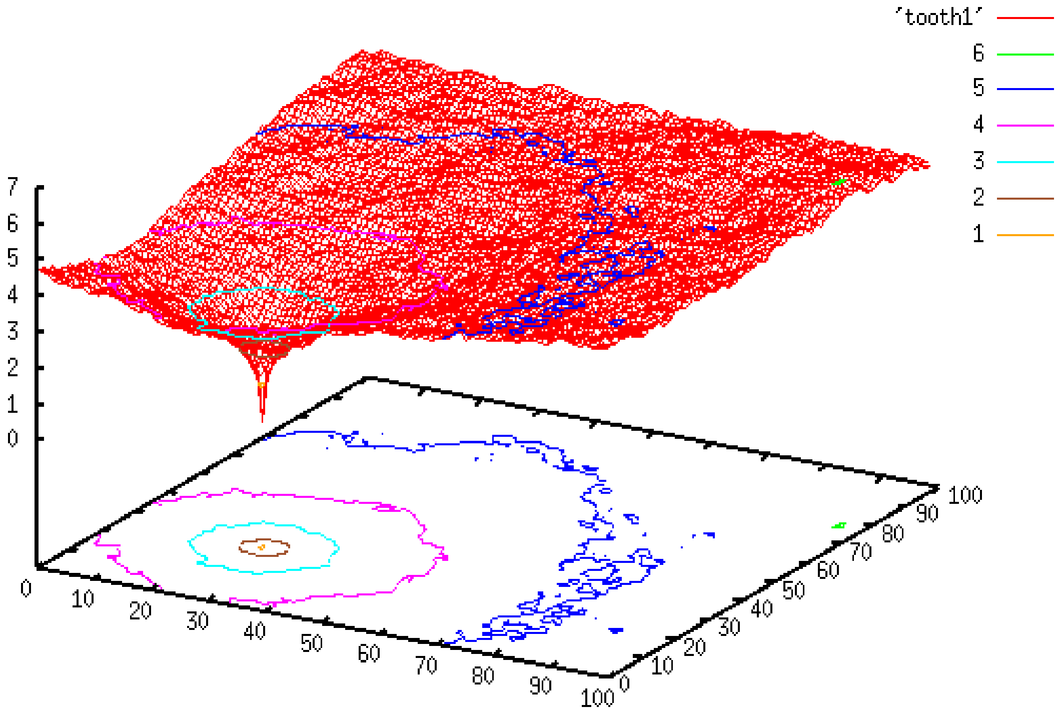

Figure 8. Embedding continuous geometries using string corobs and toothpicks

3.8 Mutual Exclusion and Concurrency

Corob tokens are stored in N-dimensional subspaces just like conventional classical tokens can be assigned a binary number and stored in a conventional n-bit register or memory. For example, the three tokens Red = 01_2, Green = 10_2 and Blue = 11_2 could be stored in a 2 bit register, which has the maximum capacity of 2^2 = 4 possible distinct binary tokens. These distinct values are *mutually exclusive* because there is room for only one binary value at a time. Alternatively, the three tokens could be orthogonally coded as Red = 001_2, Green = 010_2 and Blue = 100_2 in a 3 bit register with a maximum capacity of 2^3 = 8 distinct states, which then *simultaneously* allows any valid combination of Red, Green and Blue such as Purple = 101_2 and White = 111_2. Unfortunately, no other color shades can be represented in this encoding unless more intensity bits are allocated for each color (from 1 bit to 8 or 24 bits as used in modern computer graphics).

This discussion demonstrates that states (Red, Green or Blue) can share the same minimalistic bit encodings only if they are mutual exclusive. If multiple states must be simultaneously present then the encoding cannot be shared and must be redundant (orthogonal or separate bits allocated for each color). This same argument is true for tokens defined in subspaces in that each corob token is encoded as a random address in the subspace. The difference between subspaces and registers is subspaces naturally have extra room to simultaneously represent mixtures of naturally orthonormal tokens using the principles of vector addition.

For example, if Red, Green and Blue are each assigned a random corob in the same subspace (size N = 20 gives statistically reasonable separation) then they can be thought of as orthonormal states with an error correction region around each point. It is also possible to create points that are partway between Red and Blue (just average their addresses) and form a color we know as Purple. So even though the colors Purple and Yellow etc were not predefined for the “color subspace” they can be represented in the system as a concurrent mixture of other primary tokens.

This kind of concurrency or token mixtures does not emerge in the traditional mutual exclusive binary encodings of tokens unless they are separately coded. Because of the minimal encoding, conventional binary encoding have abrupt edges while corobs represent soft tokens with a unique kind of concurrency encoding possible. This concurrency can occur because corob points can also be thought of as orthonormal basis vectors. This concurrency is also similar to the mixture of spin states in quantum computing, thereby leading the qubit property we call superposition. The primary idea behind superposition is that the spin states are NOT mutually exclusive, so the system state can simultaneously represent both Up and Down or both True and False.

3.9 Noise Insensitivity and Reliability

Most tokens are assigned a random address in a subspace, which are naturally far apart. If some process creates a noisy version of one of the tokens (drifting away in a random direction by mixing in another unknown random token) the distance metric will still identify the closest original token compared to all the other known tokens up to the standard distance limit. This probabilistic geometry property leads to reliability of identifying tokens in the face of noise less than the amount equivalent to pure randomness. This idea that the amount of noise is directly quantifiable based on the standard distance metric is an important idea central to corob theory.

4. Distance and Information Content

4.1 Distance Metric is Information

Any probabilistic process defines a mean and standard deviation. In addition, the confidence interval is proportional to the standard deviation and can be computed using the error function (by assuming the distribution as a Gaussian). This means that a distance metric represents a probability, which can be directly converted to the equivalent information content in bits. Therefore two points at the standard distance have little information content since they are at the expected distance. Alternatively, two points that are close compared to standard distance represent a much higher information content since it is unexpected due to random processes.

4.2 Information content of each dimension

The information content of each dimension can be computed by starting with the standard distance represented as the number of standard deviations $\sqrt{N/6}/\sqrt{7/120} = \sqrt{20N/7}$, which leads to a probability using the error function prob = erfcc($\sqrt{20N/7}$. The final bit content of this standard distance probability is: bits = $-\ln_2$(prob). Figure 9 shows the plot of these equations for N from 1 through 250, where after that the error function (which is an integral) underflows for double precision floating-point numbers.

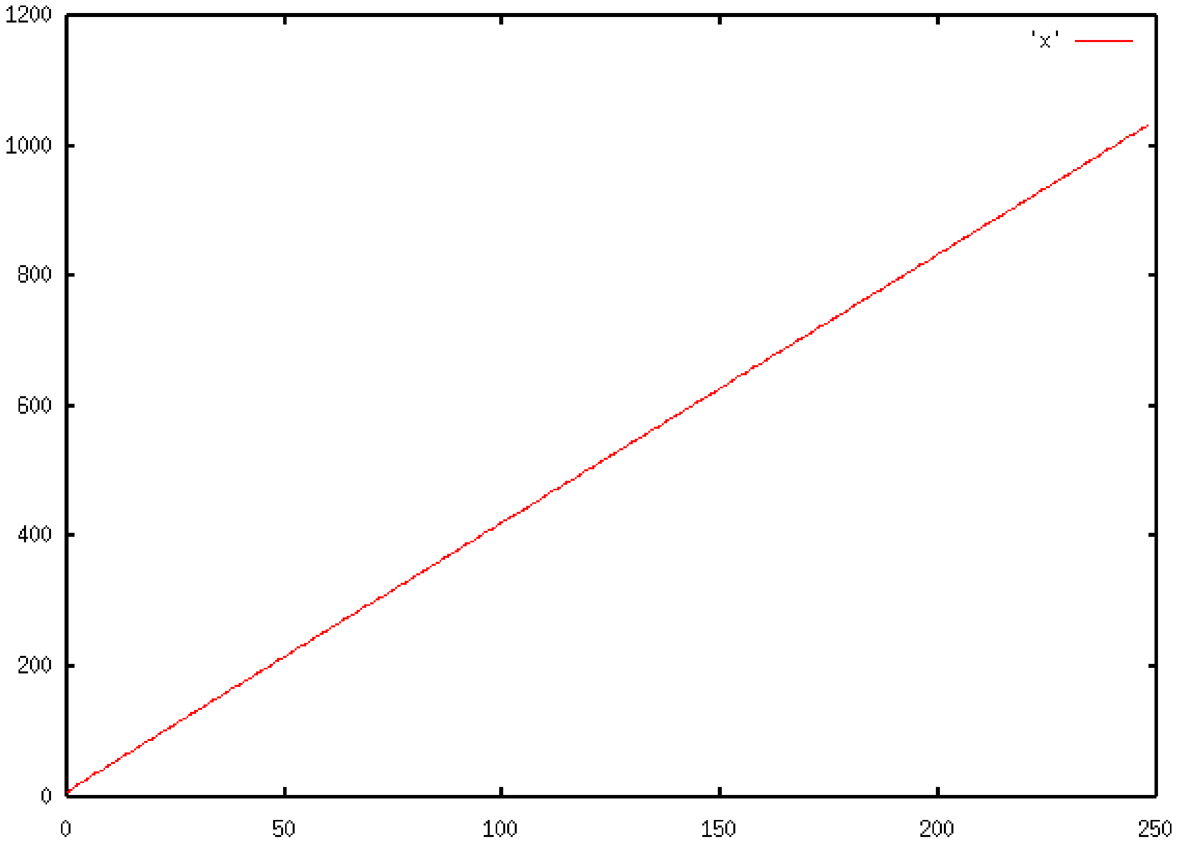

Figure 9. Bit content of standard distance for N=1-250

This plot produces a straight line that approximates the equation of $1+\left(N/\sqrt{7/120}\right)=$ $=1+(4.14*N)$. So each dimension (a floating point number $0 \le x_i \le 1$) is worth slightly more than 4 bits of information or roughly sixteen discrete values in each dimension. This makes sense since the N-space volume and the mean standard distance grows with N, so more bits are required to represent probability in larger spaces since they are effectively N-shell ratios. This limit to the precision needed for representing each dimension is primarily due to the distance metric, which sums many such squared dimensions resulting in the lower order bits becoming insignificant. This also means that the hardware accelerators for this kind of system may be able to use less than 6-8 bits of precision for each adder or multiplier used in the squaring operations.

Based on Figure 9 and the equation $1+N\sqrt{120/7}$ (or approximately $1+N(1+p)$) it is possible to represent either probabilities or bit content by creating specifically sized spaces to hold that information content. Pre-specified grid points can be carefully located in that space and then a test point can be created with multiple *simultaneous probabilities*. This strong relationship between *concurrent probabilities*, information content and distance metric is expected to be useful in the quantum-computing arena.

5. Quantum Corobs

The key step in applying corob theory to quantum computing is to compute the standard metrics for bounded, complex-valued spaces. The first step is to make the space symmetric around zero by choosing the bounds to be –1 to +1 thereby making the side length of S=2 and the space center the point [0,0,0,…]. All distance metrics normalized by the standard radius are the same due to axis relabeling. The next step converts real to complex numbers by transforming uniformly distributed random values to uniformly distributed random phases. Finally, we must deal with the fact that the vector elements are complex numbers. So the difference between complex numbers produces $z_i=(x_i-y_i)$, which can be used in Cartesian distance formula of $\sqrt{{z_1}^2+{z_2}^2+...+{z_N}^2}$. This requires the complex square of $|z_i|^2=z_i^* z_i$, where $z_i^*=a_i-ib_i$ is the complex conjugate of $z_i=a_i+ib_i$ resulting in $|z_i|^2=(a_i-ib_i)(a_i+ib_i)=a_i^2+b_i^2$, which is the scalar magnitude squared of the complex number.

When a large array of N uniformly distributed phase complex numbers with magnitude = 1 is placed through discovery of the standard metrics, the results are shown in Figure 10. This array of N complex numbers also exhibits the corob standard distance metrics. Notice how the corner distance distribution is different due to the constraint that each complex magnitude = 1. In any case the standard distance is the same after renormalization using the standard radius. Arrays of complex numbers form the essential mathematics of quantum Hilbert spaces.

Both the real and complex spaces that are symmetric around point [0,0,0,…] have the property that randomly chosen points also naturally represent a vector from the center, without any fix up. This is useful since the same representation can be used for distance metrics or vector computations (i.e. inner product). Also the center does not shift if the bounds of the space grows.

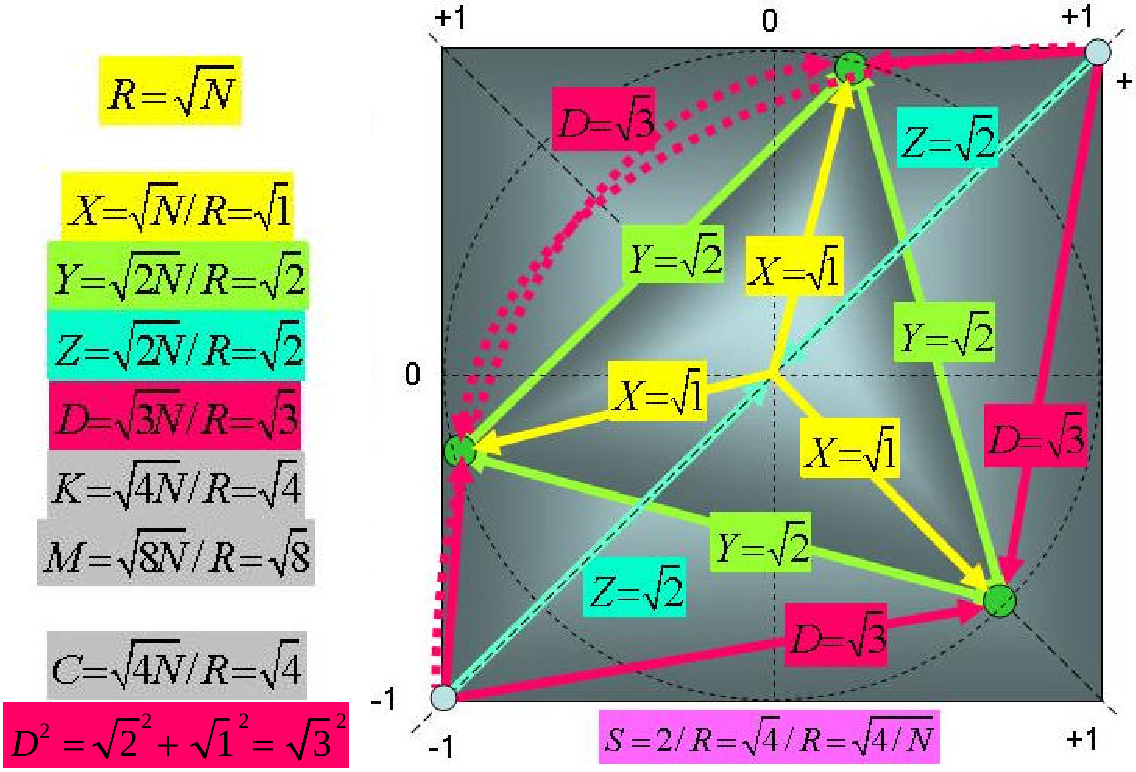

Figure 10. Uniformly distributed phase and magnitude = 1 complex numbers for N_c

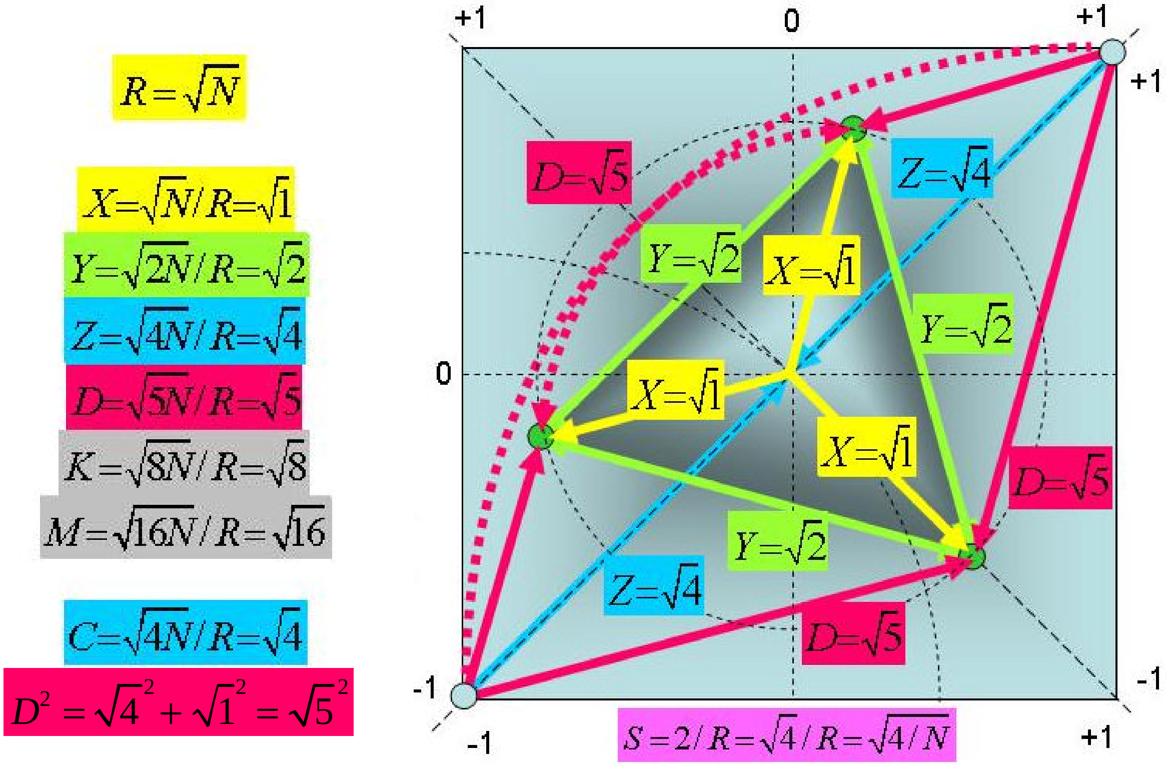

Figure 11. Uniformly distributed phase qubits for N_q or ebits for N_e with q=2

This same standard metrics discovery process can also be applied to arrays of mathematical objects of higher grade, such as isolated arrays of qubits and arrays of ebits. Figure 11 shows the same normalized distance metrics that result for both. These arrays of qubits are not the typical entangled quantum registers but simple arrays of *isolated qubits*, which we call a quantum ensemble. The arrays of ebits are also arrays of isolated ebits, where each ebit is formed by inseparably entangling two qubits. Quantum corobs are quantum ensembles formed on top of simple arrays of qubits or ebits, where the phase is the primary random variable.

5.1 Measurement as Noise Injection

It is possible to encode classical corobs into quantum corobs by converting random values to random phases. It is also possible to convert a quantum corob back into a classical corob or Kanerva bit array by *measuring* the quantum corob. The key result of this measurement process is the quantum corob survives this process. We believe this result is so significant that we have already filed a provisional US patent on the idea. Here are the details.

The key concept about this measurement process is when evenly distributed phase angles are projected; they produce results that look like a noisy version of some average result. Most importantly, the noise is smaller than the amount that would make it appear like another *completely* random point. In simple corob terms, the distance between repeated projections of a repeatable corob soft token X is smaller than standard distance, whereas the distance between projection of soft token X compared to that of soft token Y is at appropriate standard distance. Figure 12 pictorially represents these relative distances of repeatable projections.

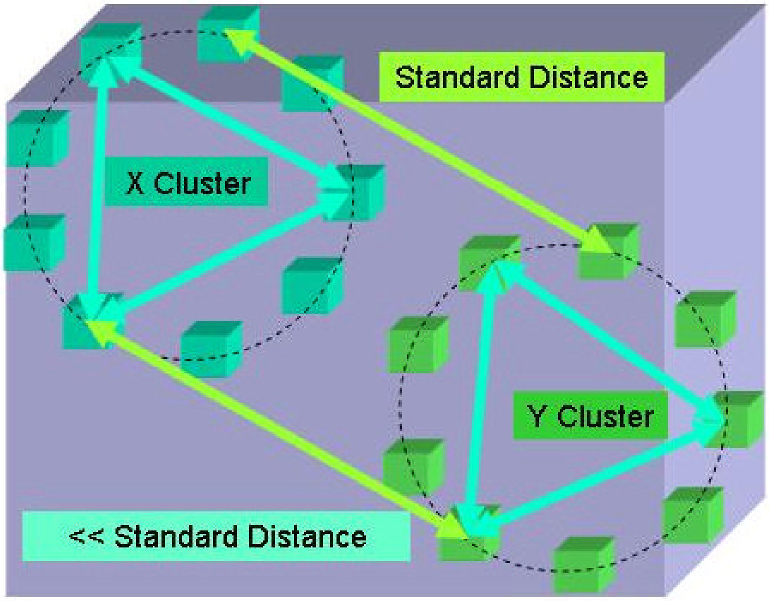

Figure 12. Cluster Projection distance for qubit arrays

Figure 13 shows the histogram that proves the earlier statement that repeated projections from a repeatable qubit soft token X acts like noisy version of some average expectation result because the distance is roughly 0.5 of standard distance. The rule of thumb we use is if the standard distance is 0.5 then the number of non-noisy dimensions is $(0.5)^2$ = 0.25 to give the same distance, so roughly 25% of the dimensions are noisy. This is possible even with arrays of random phase qubits because the projection process favors points/probabilities that are closer to either measurement basis set compared to the purely random results from phases 45 degrees away. Since the initial phase is totally random the choice of basis states is also immaterial. The histogram demonstrates that the standard deviation also shrinks as N becomes larger. The labels mean; ic = “in cluster” and bc = “between cluster”. These histograms indicate it is highly unlikely for expected values around 0.5 standard distance will ever be confused with points at standard distance, therefore repeatable, qubit-encoded quantum corobs survive measurement.

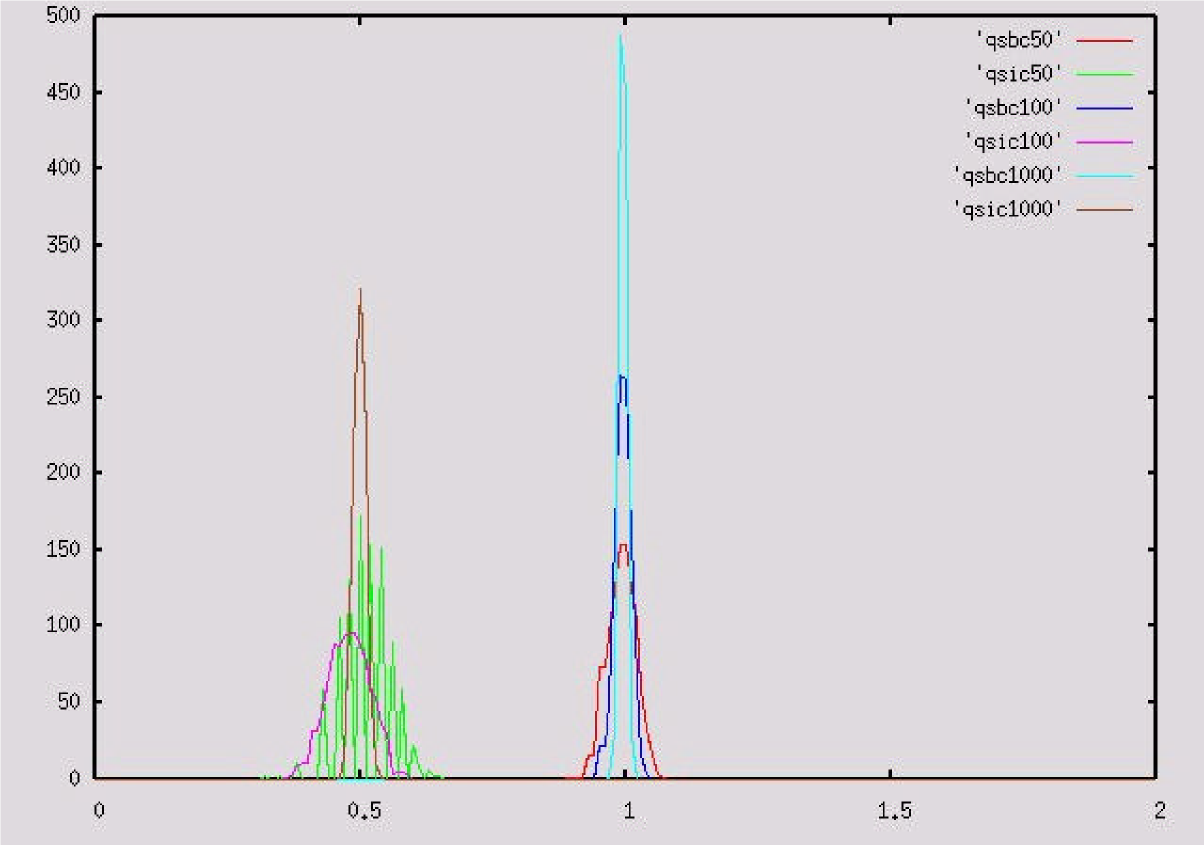

Figure 13. State cluster measurement histogram for qubit arrays N=50, 100, 1000

This result is interesting and quite unintuitive, because quantum randomness is not considered to be a computational resource. The key to this result is that isolated quantum ensembles must be used instead of entangled quantum registers. Notice discrete states when N=50 is low.

This result can be easily validated by looking at arrays of larger grain-size mathematical objects vs. complex numbers, qubits or ebits. Assume that each array element is an entangled quantum register of Q=2, 3, 4, or more qubits. Using the tensor product to create entangled qubits forms larger sets of unique basis vectors. Table 3 shows the result of taking a measurement using an arbitrary phase basis vector. The key conclusion is that the larger register the more random the measurement process produces. Even though quantum registers are producing corobs, they *generate random corobs* rather than tokens that survive. These numbers are slightly different than the talk slides due to improvements in our libraries of generators.

Table 3. State Cluster measurement distances for arrays of higher grained qubit registers

Element	% of standist	# of reals	Entangled state equation
complex	70.7 %	2	a (is complex)
1 qubit	50 %	4	$a\|0\rangle + b\|1\rangle$
2 qubits	65 %	8	$a\|00_2\rangle + b\|01_2\rangle + c\|10_2\rangle + d\|11_2\rangle$
3 qubits	76 %	12	$c_0\|000_2\rangle + c_1\|001_2\rangle + \ldots + c_6\|110_2\rangle + c_7\|111_2\rangle$
4 qubits	83 %	16	$c_0\|0000_2\rangle + c_1\|0001_2\rangle + \ldots + c_{14}\|1110_2\rangle + c_{15}\|1111_2\rangle$
>4qubits	100 %	>16	$c_0\|0\rangle + c_1\|1\rangle + \ldots + c_{N-2}\|N-2\rangle + c_{N-1}\|N-1\rangle$

This same result is illustrated graphically in Figure 14 for a fixed N=1000 with changing grain size of q=1 vs. q=2. These measurements are the cluster distances of the new state results after measurement.

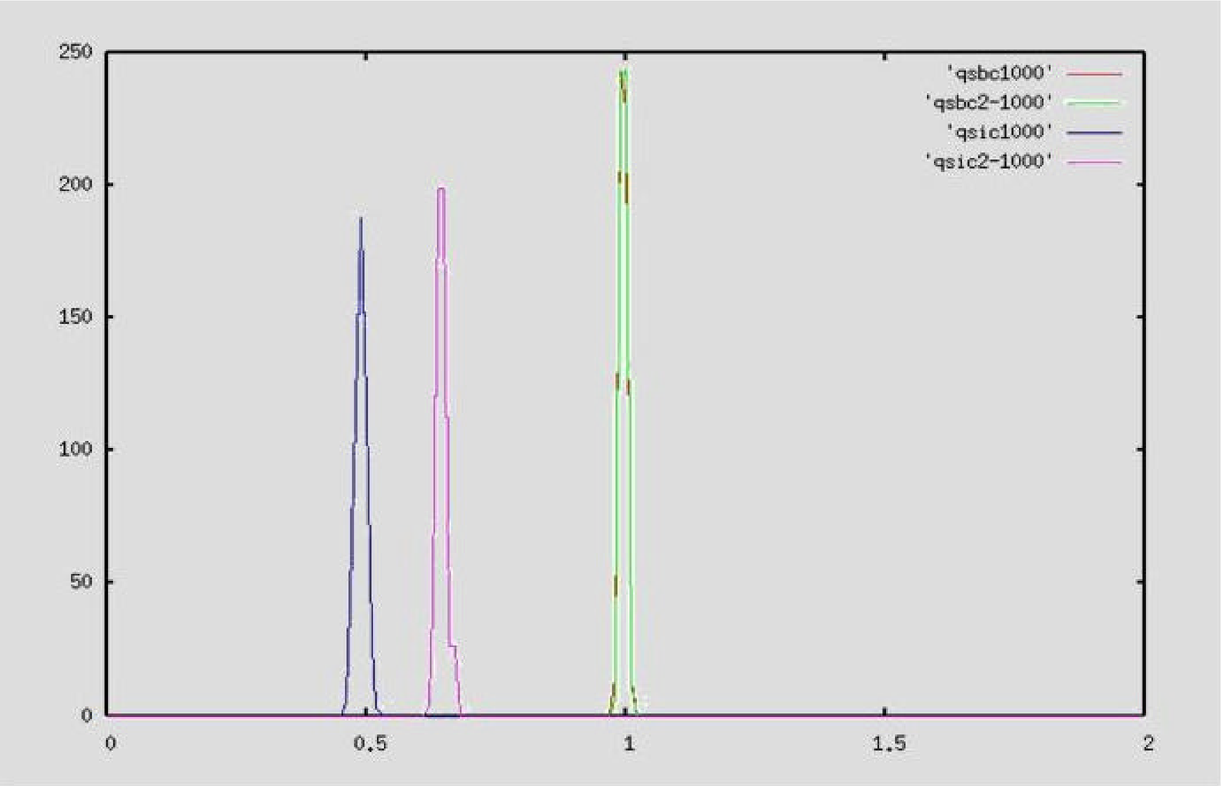

Figure 14. State Cluster histograms for arrays N=1000 of qubit registers with q=1 and q=2

The new resulting state is only part of the measurement process. The *answer* can also be viewed as a corob state since it is a binary valued Kanerva point. If many answers are averaged together they form the expected value, which is just the probability of finding a 1, so can be represented as real-valued corob. Figure 15 shows the histogram distributions of the measurement answers.

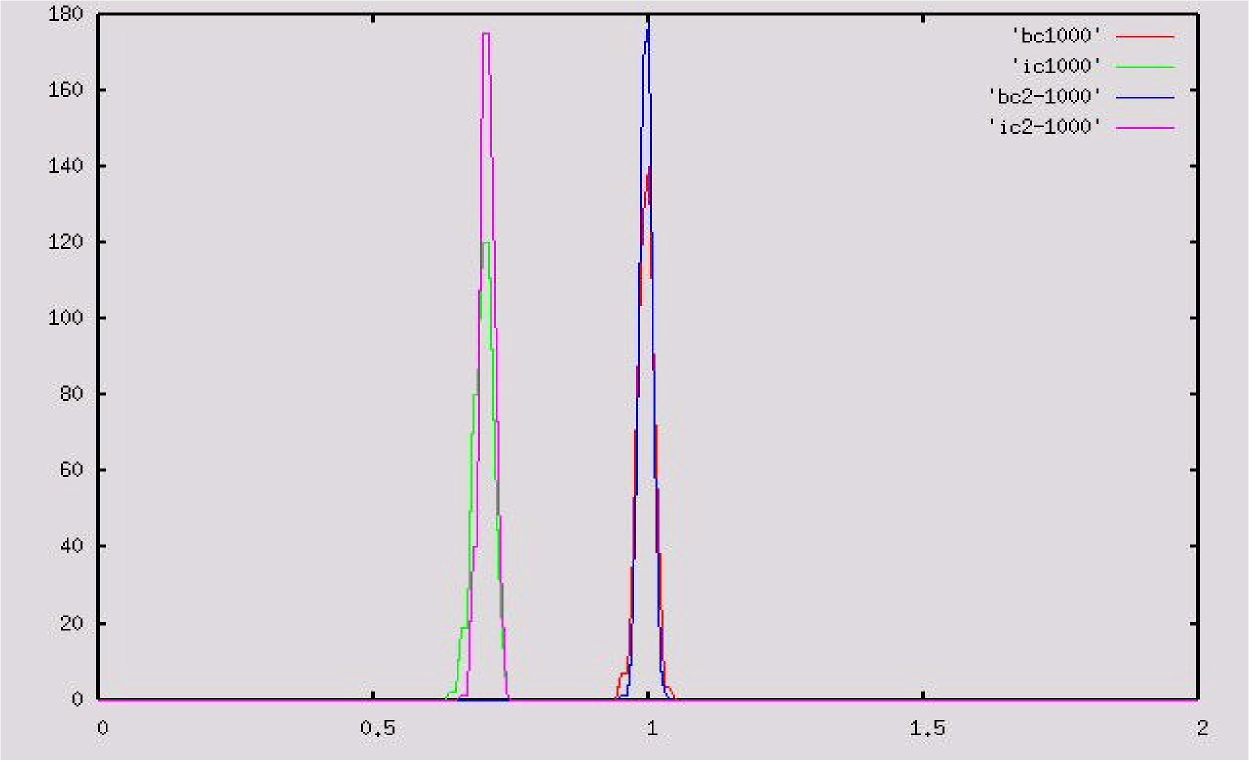

Figure 15. Answer cluster histograms for arrays N=500 of qubit registers with q=1 and q=2

In conclusion, isolated arrays of qubits (or quantum ensembles) allow representing quantum corobs that survive measurement, whereas quantum registers produce random corobs.

5.2 Ebit corobs survive measurement

Constructing arrays of uniformly distributed random phase complex numbers or qubits (pairs of complex numbers) is mathematically straightforward. A qubit is represented as expression $a|0\rangle + b|1\rangle$ where complex numbers *a* and *b* must meet the unitarity constraint of $a^2 + b^2 = 1$. The first step in investigating corobs using ebits is that generating uniformly distributed arrays of random phase ebits is not as simple. These ebit states are the *inseparable entanglement* of two individual qubits. With complex numbers *a*, *b*, *c* and *d* an ebit has the representation of $a|00\rangle + d|11\rangle$ (or $b|01\rangle + c|10\rangle$) with respective unitarity constraints of $a^2 + d^2 = 1$ with $b = c = 0$ (or $b^2 + c^2 = 1$ with $a = d = 0$), which both meet the overall constraint of $a^2 + b^2 + c^2 + d^2 = 1$. Notice how no tensor product of two smaller states can produce either combination of states, which is the definition of inseparability for ebits. It is still possible to mathematically produce a uniform random phase distribution of ebits after the tensor product and erasure steps.

When an ebit is measured, the two qubits are then broken apart into two separate unentangled qubits of the appropriate states. So the measurement for an ebit array of size N produces two qubit arrays of size N and a binary answer array of size N. These can all be analyzed from a standard distance metrics with the following results. Again Figures 16 and 17 are slightly different than the presented slides due to improvements in the ebit modeling libraries.

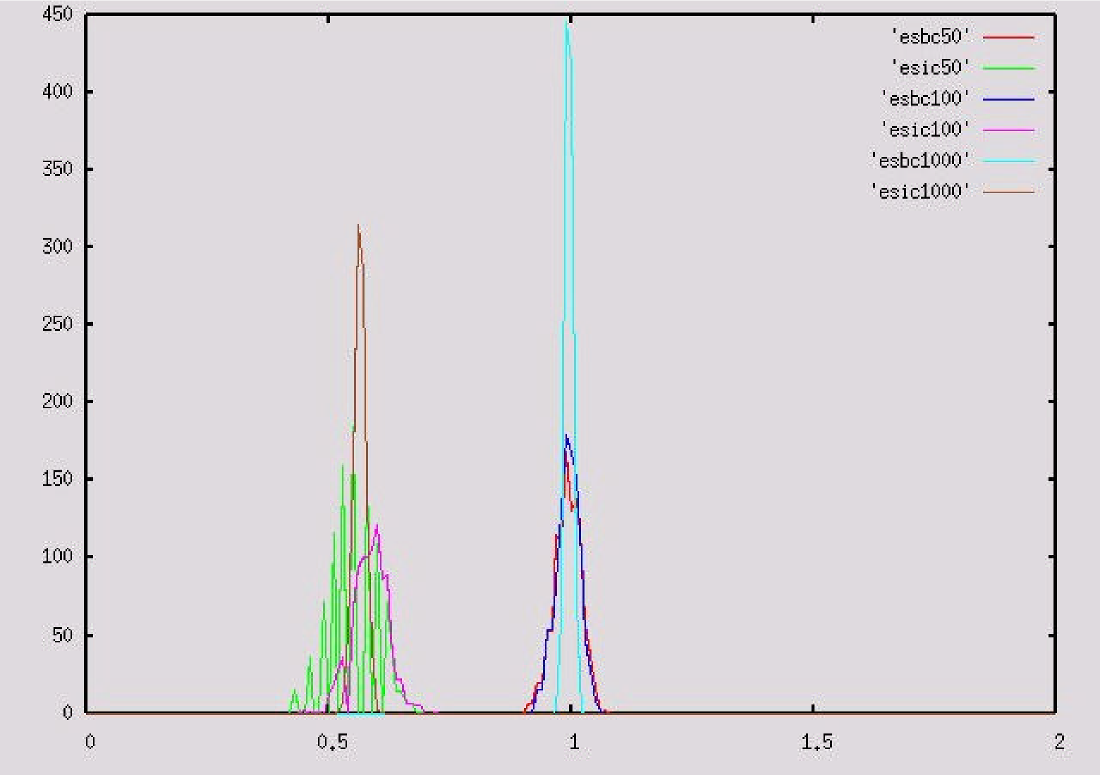

Figure 16. State cluster distance histogram for ebit arrays (q=2)

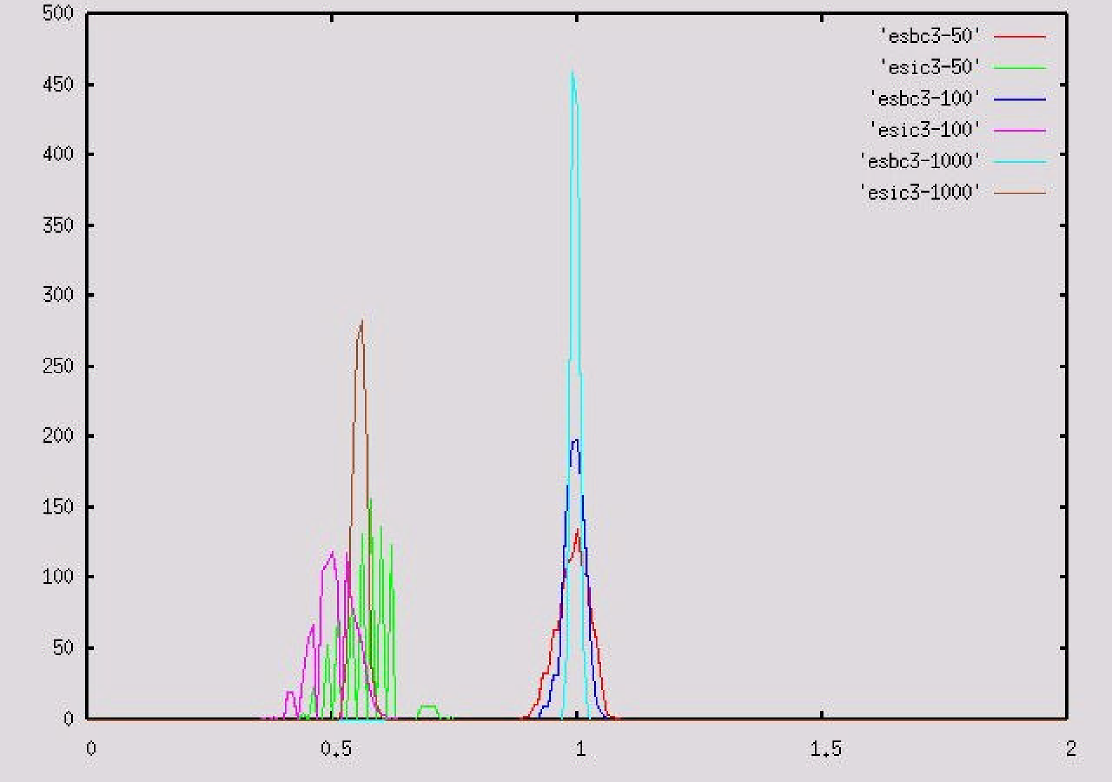

Figure 17. State cluster distance histogram for ebit arrays (q=3)

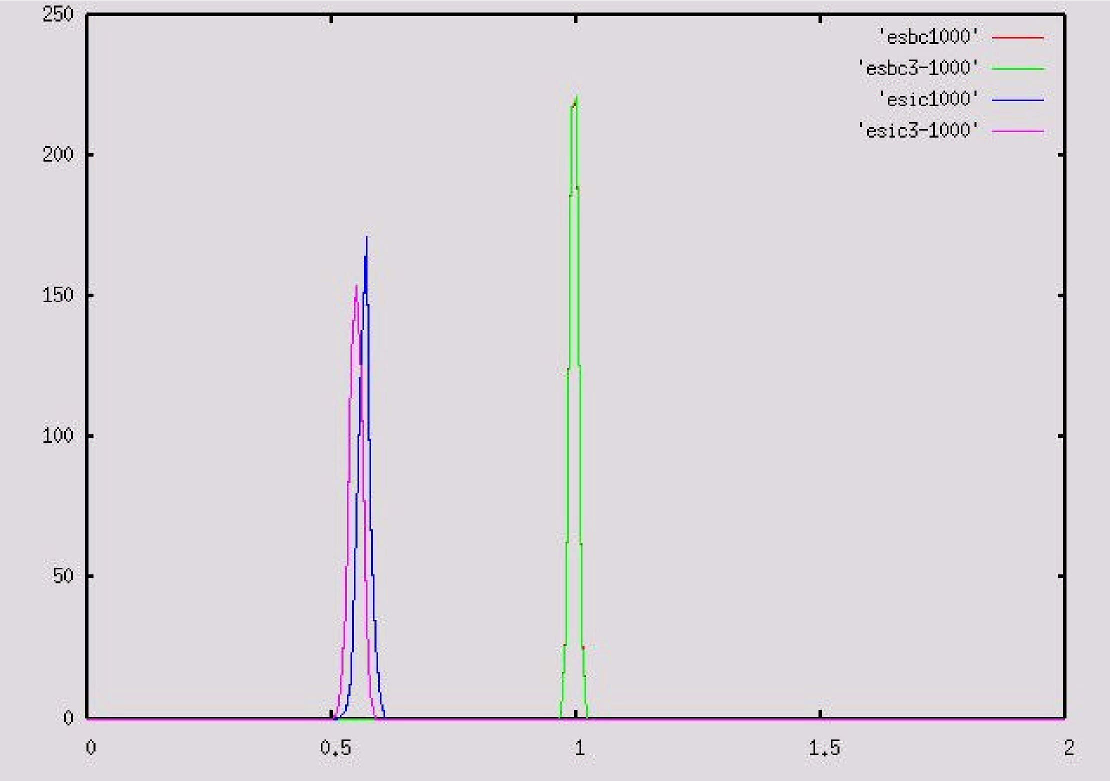

Figure 18. State cluster distances for ebit arrays N=1000 with q=2 and q=3

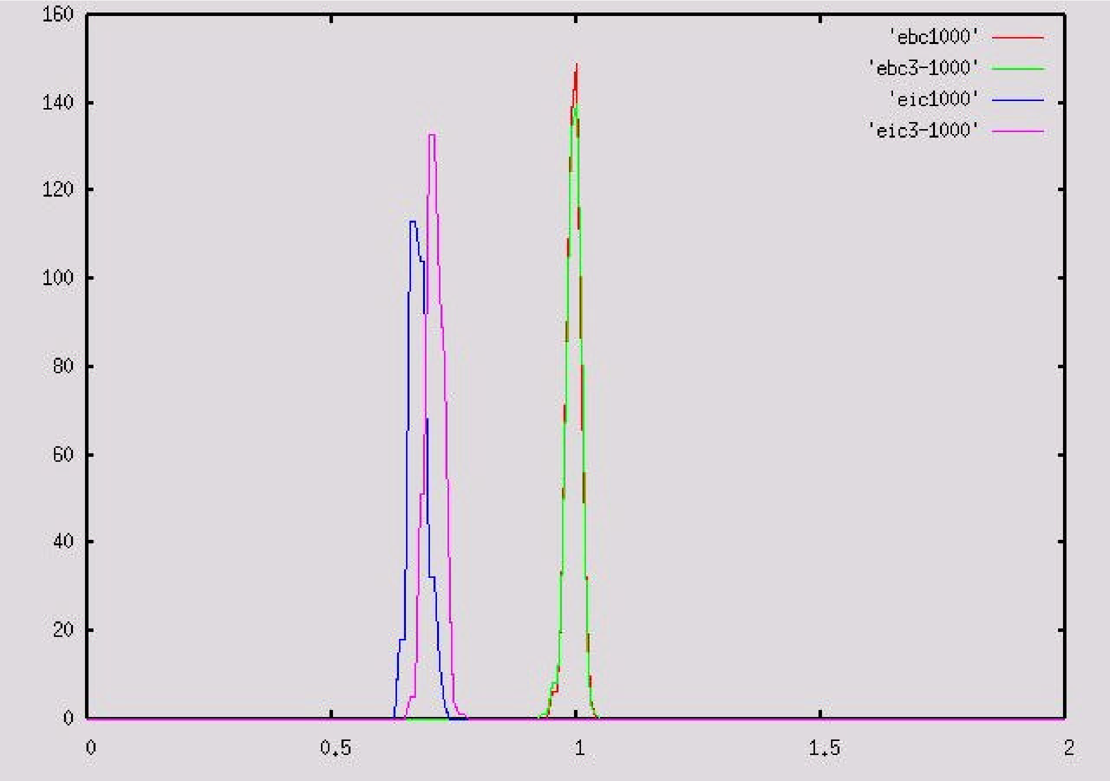

Figure 19. Answer cluster distances for ebit arrays N=1000 with q=2 and q=3

The overall result of measurement of quantum corobs represented in ebit arrays, is the both the answer and the qubit results indicate survival of the tokens. Quantum corobs survive measurement. Measurement of highly entangled quantum registers produce random corobs.

6. Conclusions

The striking result of this research is that noise and quantum randomness can be quantitatively viewed from the corob perspective. This result shows that unentangled quantum ensembles can be useful for representing unique data tokens in both the classical and quantum domains. Most importantly, these quantum data tokens can survive quantum measurement, which may be useful for any neurological theory that would make use of quantum mechanisms.

In order to explore this interrelation between classical and quantum corobs, we are building a corob language and tool suite that would allow systems to be defined containing arrays of both types of cells. This language is currently going thru the second design revision. The key computational mechanism is lightweight concurrent rules that act as unique observer perspectives defined by corobs or basis states in various subspaces. These rules appear to have the characteristics of quantum superposition because states can evolve in multiple simultaneous paths. It is expected that corob language with quantum extensions would allow research in interesting quantum domains such as quantum key distribution and quantum computation.

These results are novel and interesting since most people researching quantum computing do not know about corob theory. Anytime a large number of independent random values are represented by a system then standard distance metrics emerge. Program universe also has these same properties so therefore corob distance metrics must naturally emerge from that endeavor. The corob theory also is quite instructive about how stability can emerge from purely random processes (since the discrimination metric is identical to distance metric), which is virtually the same goal as Program Universe.

References

1. D. Klain, G. Rota, Introduction to Geometric Probability, Cambridge University Press, 1999. Also see http://pauillac.inria.fr/algo/bsolve/constant/geom/geom.html

2. P. Kanerva, Sparse Distributed Memory, MIT Press.